Robust Control of Diesel Drivelines

Abdolreza Fallahi

A thesis submitted in partial fulfillment of the requirements of the University of East London for the degree of

Doctor of Philosophy

May 2012

Acknowledgements

This research was inspired by my supervisor, Professor Stephen Dodds, whose valuable technical support enabled its successful completion. This was complemented by much encouragement on a personal level from Professor Dodds and his wife, Margaret, without whom the journey would have been more difficult. I would also like to thank my colleagues, A. Rawat, J. Pedersen and S. Kanchanavally, who also provided moral support. I thank Dr W. Hosny for being my second supervisor. Also, I would like to take this opportunity to thank my managers G. Forward, D. Higham and A. Potter. I owe many thanks to Janet (my wife) and children.

Summary

The original contribution of this thesis is to provide insight into research of non-traditional control techniques for automotive power train applications, culminating in experimental evidence of much improved performance and reduced commissioning costs. This includes much work on the technique of Observer Based Robust Control (OBRC) which, before the research documented in this thesis commenced, was only in its infancy with some promise being shown through the simulation of electric drive applications (Dodds, 2007). The thesis, therefore, contributes to the process of bringing this new control technique nearer maturity. OBRC is based on an observer designed to provide information enabling effective control of an automotive power train application and its performance assessment. Comparison with traditional and other robust control techniques is included. The observer in OBRC is is designed to estimate the equivalent disturbance input, referred to the control input to a plant. This represents plant modelling errors as well as external disturbances. The equivalent disturbance estimate is applied to the real plant input to cancel its effect, thereby reducing the control problem to that of controlling the known real-time model of the plant employed in the observer. One of the disadvantages of conventional robust control methods, such as those based on sliding mode control, is that relatively high gain control loops are closed around the uncertain plant. This increases the risk of instability due to the dynamic elements, such as sensor lags, that are not included in the assumed plant model. The initial reason for investigating OBRC is that the high gain loops are applied to the known plant model in the observer and that the stability of these loops, taken in isolation, can therefore be guaranteed. It was found that the observer gains are limited only by the finite sampling frequency of the digital processor. In theory, infinite observer gains would yield ideal robustness. However, in practice only finite gains are possible. The aforementioned application of the equivalent disturbance estimate to the real plant input effectively transfers the high gain loops from the plant model in the observer to the real plant. This meant that closed loop stability could not be guaranteed under all circumstances. In view of this, it was decided that sliding mode control should not be excluded from the

set of controllers for comparison. Since the control chatter associated with basic sliding mode control has to be eliminated for the vehicle application, polynomial control (a continuous version of the discrete RST controller) with robust pole assignment is included. This polynomial control can be regarded as equivalent to the sliding mode control with a boundary layer, but without the uncertainty associated with the choice of the boundary layer width. Two more controllers, based on the Internal Model Control (IMC) and H-infinity, are included for comparison on the basis that their design methodologies do not demand high gains. The various control techniques are demonstrated and compared via their application to Diesel Drivelines for commercial road vehicles.

One of the operational problems with conventional PI engine speed controllers is the need for time consuming initial controller tuning. This requires different sets of gains for each gear selection, including idle (i.e., neutral) and later retuning to compensate for changes in the driveline characteristics with component aging. A major advantage of OBRC in this application is the elimination of the tuning procedure. Of particular interest is the fact that the order of the system is increased by two when a gear is engaged due to a vibration mode created by the finite torsional compliance of the propeller shaft and other driveline components. Since this driven mechanical load can be represented by its inverse dynamic model in a feedback path whose output acts at the same point as the control variable, the OBRC compensates for this automatically without the need for any parametric changes. The simulations and experimental work were carried out on the DAF 12 litre diesel engine. The comparative study was carried out, not only with respect to the main application of Diesel Drivelines, but also using academic examples that are even more demanding of the controllers' capabilities.

A key parameter in an engine configuration is the saturation limit on the injected fuel rate, which is highly dependent on the engine capacity from 10 litres to 16 litres. One example was introduced with a saturation block and the abilities of the various controllers under this constraint were assessed. The H-infinity controller could not handle such a saturation constraint, which is common practice in automotive

applications and therefore this had to deemed unsuitable. The remaining controllers were able to operate with fuel rate saturation.

The overall conclusion is that the controllers based on OBRC, polynomial control and IMC are capable of a similar performance with appropriate controller parameter settings. However all are subject to the trade-off between the conflicting requirements of short response times and robustness.

Contents

List of Symbols

P	Plant Transfer Function
G	Plant Transfer Function
^	This symbol represents an Estimated Value
z^{-1}	Time delay of One Sampling Period
K(s)	Compensator
C(s)	Compensator
S(s)	Sensitivity Function
T(s)	Complementary Sensitivity Function
n	Positive Integer Indexing (0,1,…,n,)
u	Control Signal
T_{co}	Time Constant
T_s	Settling Time
s	Complex Variable of the Laplace Transforms
l_r	Road Load e.g. hills
l_v	Vehicle Load
l_{rv}	Sum of the l_r and l_v
ω_b	Bandwidth of System
τ_{fd}	Filter Time Constant for Derivative

List of Acronyms

OBRC	Observer Based Robust Control
IMC	Internal Model Control
EMS	Engine Management System
SISO	Single Input Single Output
RHF	Transfer Function Blocks, R(s), H(s) and F(s) in continuous polynomial control structure
RST	Transfer Function Blocks, R(z), S(z) and T(z) in discrete polynomial control structure
rpm	Revolutions Per Minute
rad	Radian
PID	Proportional, Integral, Derivative
CAN	Controller Area Network
PTO	Power Take Off
LHP	Left Half Plane
RHP	Right Half Plane
SSE	Steady State Error
ETC	Electronic Traction Control
ABS	Anti-lock Brake System
y_{inv}	Inverse dynamics representation of vehicle model
y_{std}	Vehicle model in standard form

List of Figures

1 Introduction

1.1 Motivation for Research

The aim of this research is to investigate control techniques that, in contrast with traditional ones, do not require time consuming tuning procedures at commissioning time together with subsequent and frequent retuning due to plant component ageing, and to make recommendations for the future based on performance comparisons by simulation and physical tests on an engine. A practical constraint is that the new control algorithms can be implemented on currently available fixed point real time embedded Micro-controllers with limited computation power. The investigation includes comparison with presently implemented control techniques.

Figure 1-1 shows a typical vehicle upon which the control techniques under investigation might be implemented.

Figure 1-1: Tractor of the Articulated Vehicle

The diesel engine controllers in the heavy duty sector (typically 10ltr to 16ltr) have been controlled by 16 bit and 32 bit real time microprocessors, with fixed point capability only. There is, however, a slowly developing trend towards floating point capability in the new generation of microprocessors. The existing products, however, need to be maintained and will require updates. This motivates the evolvement of improved control techniques with minimal complexity.

The existing control algorithms that are run on the current products predominantly implement PID (Proportional, Integral and Derivative) controllers. Although the PID controller is linear in its basic form, non-linearities are commonly built into the controllers for practical reasons. These nonlinearities vary in complexity, from simple saturation on the actuator input and/or integral term anti-windup, to complex gain scheduling based on the vehicle load and vehicle speed. There has been very little change in the industry, as 95% of Engine Management Systems (EMS) are still using PID controllers but control system improvements have evolved in the domain of the gain scheduling. One reason for the PID controller continuing as the basic workhorse of engine management systems is due to its track record of working reliably in various applications, albeit with room for improvement in performance. The tuning however is time consuming and has to be repeated during control system development and often at service intervals. This is to compensate for the drift of the plant parameters, due to component wear and ageing. Different vehicles may require different gains for a given application. For example, if an application for a fire engine or a crane is used, then different gains are required for the road speed controller. Hence one has to tune the controller each time for a given application. The research programme is therefore directed towards robust control techniques that avoid tuning and retuning. This will take advantage of the digital implementation medium, particularly the future floating point processors, to reach performance levels unattainable with PID controllers. The author initially developed a road speed limiter with a PI controller [48], including nonlinear and variable gain scheduling. The design had to cater for an empty vehicle (4,000kg), a fully loaded vehicle (40,000kg) or a half empty fuel tanker truck. This also had to accommodate driveline

oscillation [51] and work with several different truck manufacturers. A DAF Truck commission led a study into the robustness of the PI controller [48].

1.2 Original Contribution

The original contribution of this thesis is the assessment of the new OBRC control technique, which is described fully in Chapter 4, together with three other control approaches comprising polynomial control, IMC, described in Chapter 7 and H-infinity, culminating in a recommendation of those that would be advantageous in future engine speed control applications. The components of this original contribution are as follows:

I: - The OBRC, which through its inherent robustness, ensures that the vehicle speed follows a transient path which is prescribed by the control performance specification, despite changes in the vehicle dynamics and external disturbances.

II: - The inverse dynamic method of estimating the model uncertainty (caused by the un-modelled component of the plant and the external disturbance).

II: - Simplifying the modelling of a vehicle by means of the inverse dynamic method and thereby reducing the required design effort in terms of man-hours, when employing OBRC or polynomial control with robust pole placement.

IV: - The introduction of OBRC and polynomial control that eliminates tuning of the controller in real time on the physical vehicle.

V:- A new numerical method for pole assignment.

VI:- A pre-filtering technique producing a prescribed transient behaviour..

1.3 Overview of Engine Control

The Diesel engine has been widely used as a power source in mass transportation. Diesel engines are used in many applications such as articulated lorries, passenger vehicles, medium duty vehicles, generator set prime movers and ship propulsion. The capacity of these engines varies from 2.0 L up to 16L for mass transportation.

Most early engine speed control systems were based on mechanical governors which were not flexible in terms of adjustable design parameters and consequently suffered from compromised performance.

It is a well-known fact that the diesel engines are highly nonlinear plants and their characteristics vary as function of power output, coolant temperature, oil pressure, oil temperature, turbo charger characteristics and many other factors. The diesel engine also renders the control system a discrete time one because the combustion occurs every120 degrees of crankshaft rotation for a 6 cylinder, four stroke engine. In view of the impulsive torque produced, the engine speed is filtered by a large flywheel. Furthermore, a diesel engine is inherently open loop marginally stable in the sense that the engine speed will drift away in the absence of closed loop control. Hence to prevent the engine from running away or stalling, it requires a governor.

In general the engine control structure varies from company to company, such as Delphi or Bosch. But they all have a basic structure which includes a strategy for starting the engine, known as cranking, and a few other engine states such as running, idling, cruising , PTO, etc.

The engine position and phasing is provided by two toothed wheels, usually mounted on the engine flywheel and cam shaft. The cam shaft wheel teeth provide phasing of the engine position and usually have N equally spaced teeth, where N is the number of cylinders, plus one extra tooth that indicates cylinder 1 is approaching its compression stroke. This wheel is known as an N+1 wheel. The flywheel teeth are more numerous, usually being placed at intervals of 6 crank degrees spacing. This wheel would therefore have 60 equally spaced teeth but 1 or 2 teeth are removed, to allow the system to run the engine if the cam shaft sensor fails.

Figure 1.2 shows a typical structure of a Diesel engine control system whose functions fall into one of the following two categories,

I: - Vehicle Control: Via the torque demanded by the driver (pedal, cruise control via user interface, etc.) or via a CAN link which uses the J1939 standard, and allows other devices (ABS, ETC, etc.) on the vehicle to control or limit the engine torque.

II:- Engine Protection: A torque limit protects the engine against mechanical and thermal damage. The full torque curve will protect the engine at the lower end of the engine speed from mechanical damage and provides thermal protection at the upper end. The maximum engine speed governor will protect the engine from mechanical damage due to valve bounce and excessive internal forces and torques due to component accelerations.

The smoke limiter prevents the engine from producing smoke due to incomplete combustion by limiting the fuel quantity injected into the cylinder. The smoke limiter becomes active during transient (change in load or step in speed demand,) and remains active until the turbo charger catches up and provides the required air to the engine to achieve complete combustion. During steady state operation, there is excess air in the system, and therefore the smoke limiter is not active.

The idle speed governor prevents the engine from stalling and maintains the engine speed when any auxiliary (power steering, air heater, etc) demands extra power.

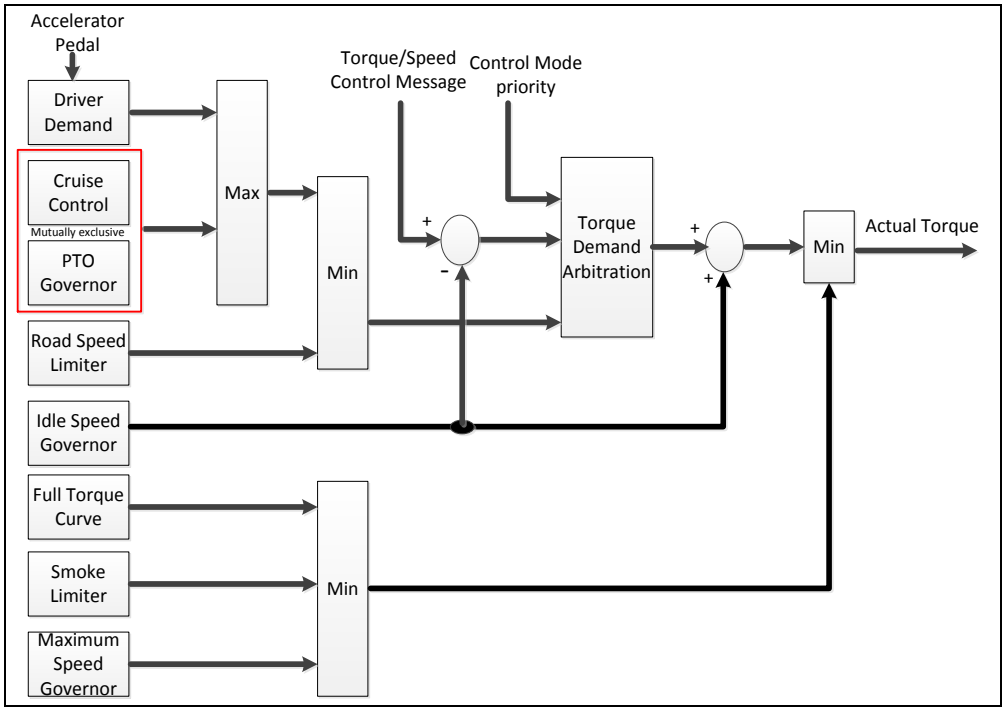


Figure 1-2: Overview of a Diesel engine control structure

Figure 1.3 provides an overview of a typical engine operating envelope related to the accelerator pedal position. Typical a pedal torque demand is a function of engine speed and pedal position and is realised by means of a 3D lookup table. The lookup table is used to create a proportional governor. It uses the line of constant pedal position against engine speed and will reduce the pedal torque demand as the engine speed increases. Some other applications may use the accelerator pedal position as the engine speed demand (this is known as all speed governors) and this becomes a reference input to the controller.

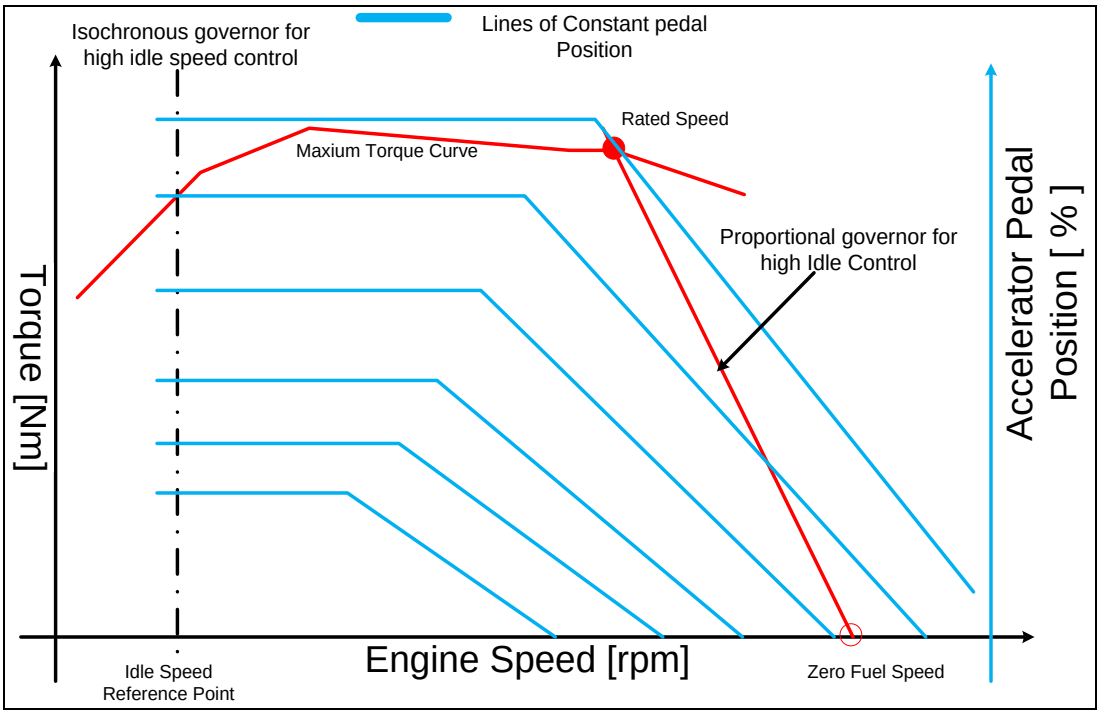


Figure 1-3: A typical working envelop of the diesel engine

1.4 Thesis structure

The thesis comprises of 6 chapters', the content of which is as follows:

Chapter 1 provides a statement of the motivation for the research, a summary of the original contributions and some background on engine management systems.

Chapter 2 provides the necessary background needed for understanding the material of the subsequent chapters. It reviews the block diagram structure of the basic control loop and its transfer function relationships, and defines the control system properties of robustness, sensitivity and internal stability, discusses control system design methodology and finally introduces the settling time formula as a simple design tool for realising transient response specifications for control systems of arbitrary order.

Chapter 3 first introduces a special form of observer for estimation of the external disturbance referred to the control input and goes on to show that in the absence of an external disturbance, this estimate enables plant parametric errors to be corrected, and furthermore that if only a portion of the plant is modelled in the observer that the disturbance estimate contains information about the remainder of the plant represented in the inverse dynamic form. The application to a Diesel driveline is considered. Finally, some preliminary experimental results and corresponding simulations are presented of control of an engine subject to electromagnetically generated load torques from a dynamometer, including the special observer and use of the load torque estimate as part of the control signal to counteract the real load torque.

Chapter 4 develops the observer based load estimation of Chapter 3 into the general OBRC robust control technique as this is one of the original contributions of this research programme to the field of engine management systems.

Chapter 5 introduces the control techniques to be compared with one another and with the OBRC of Chapter 4 and then carries out the comparisons by analysis and simulation using various plant examples.

Chapter 6 contains the comparisons of the control techniques by simulation and experiment specifically for the Diesel driveline application.

Chapter 7 presents the conclusion of the thesis and recommendations for further research.

The Appendix provides some supporting theory of the control techniques and some design software.

2 Essential Background and Definitions

2.1 Classical Control Loop Structure

Figure 2.1 shows the classical form of a linear continuous SISO closed loop system block diagram with unity feedback. The general transfer function relationship used for analysis will now be derived and definitions of terms made, where appropriate.

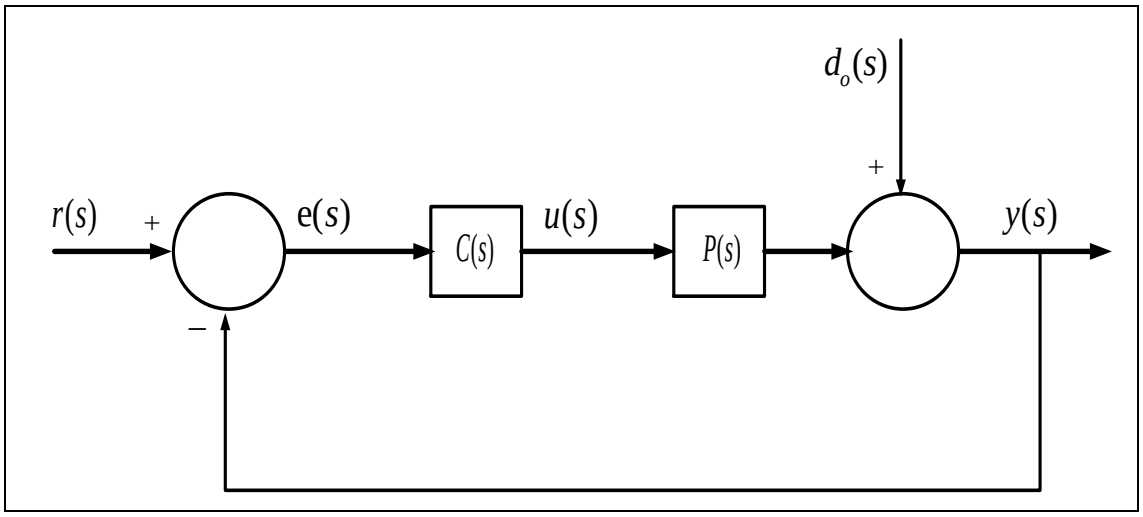


Figure 2-1: SISO Linear Feedback Control System Block Diagram
with the external disturbance referred to the output

Here are some general terms and equations that will be derived from the block diagram shown in Figure 2.1 [45]. The "error" is defined as the difference between the reference input, $r(s)$, and the plant output, $y(s)$, to be controlled. Thus

$$e(s) = r(s) - y(s) \tag{2.1}$$

This is also defined as the "controller error" and is input to the controller with transfer function, $C(s)$.

The output of the controller $u(s)$ is then given by.

$$u(s) = C(s)\mathrm{e}(s) \tag{2.2}$$

The external disturbance, $d_o(s)$, is referred to the controlled output in Figure 2-1, which is typical in the formulation of control loops designed by H_∞ methodology [9, 32]. Thus

$$y(s) = P(s)u(s) + d_o(s) \tag{2.3}$$

By substituting for $u(s)$ using (2.2) and (2.1), the closed loop transfer function relationship may be derived as follows:

$$y(s) = P(s)C(s)\big[r(s) - y(s)\big] + d_o(s) \Rightarrow \tag{2.4}$$

$$y(s)\big[1 + C(s)P(s)\big] = C(s)P(s)r(s) + d_o(s) \tag{2.5}$$

Hence, $$y(s) = \underbrace{\frac{C(s)P(s)}{1 + C(s)P(s)}}_{T(s)} r(s) + \underbrace{\frac{1}{1 + C(s)P(s)}}_{S(s)} d_o(s) \tag{2.6}$$

where *T(s)* is the closed loop transfer function and $S(s)$ is known as the sensitivity transfer function. It should be noted that this provides a measure of the effect of the external disturbance n the output. For further details, the derivation of the sensitivity can be found in Appendix I A.1.1.

The loop transfer function is defined as $L(s) = C(s)P(s)$ for the unity feedback system as shown in Figure (2-1).

By inspection of (2.6),

$$T(s) + S(s) = 1 \tag{2.7}$$

Hence $T(s)$ is sometimes referred to as the complementary sensitivity function.

2.2 Sensitivity Definition

Sensitivity is the proportional change in the closed loop transfer function divided by the proportional change in the open loop transfer function. For analysis of the robust control techniques other than H_∞, the external disturbance, $d_i(s)$, is referred to the control input, as shown in Figure 2-2.

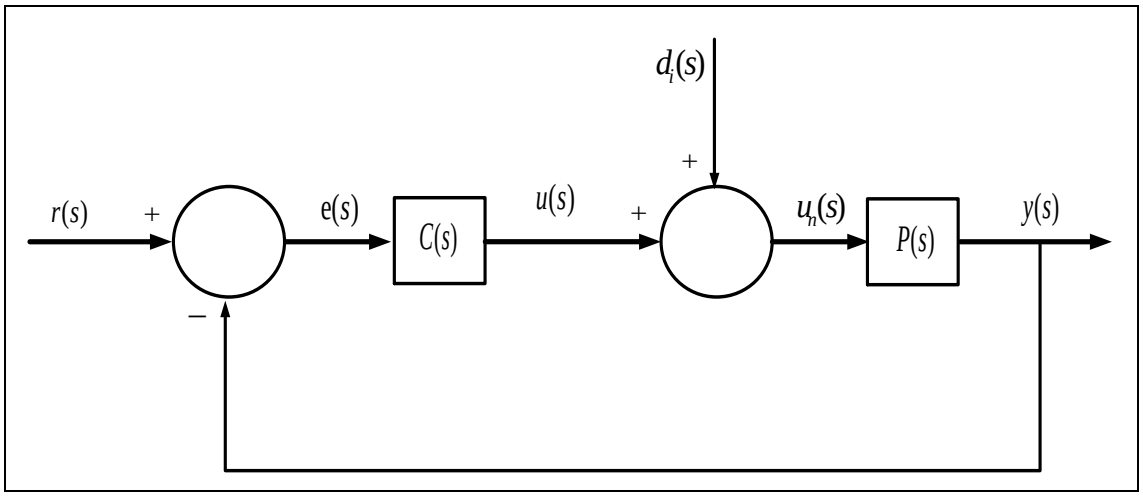


Figure 2-22-2: SISO Linear Feedback Control System Block Diagram
with the external disturbance referred to the control input

In this case, the sensitivity, $S(s)$, given in (2.6) is also the transfer function between the external disturbance, $d_i(s)$, and the net plant input, , $u_n(s)$, as shown in Figure 2.2. In the frequency domain, the sensitivity is the Bode magnitude plot, $S_{\text{dB}}(\omega) = 20\log_{10}\left|S(j\omega)\right|$. Then an upper threshold boundary, $S_{\text{dB}\max}(\omega)$, is specified for $S_{\text{dB}}(\omega)$. The specification is satisfied if $S_{\text{dB}}(\omega) > S_{\text{dB}\min}(\omega)$.

A typical control system specification including sensitivity could be as follows:

1. ω_b The minimum bandwidth criteria (disturbance rejection).
2. The maximum peak of $|S(j\omega)|$.
3. The steady state error of the closed loop system.

2.3 Robustness Definition

There are several quantitative definitions for the robustness of a control system. Each definition has an emphasis on a different aspect of the closed loop behaviour. In general, however, it is defined qualitatively as its ability to maintain the specified closed loop performance within given limits in the presence of an external disturbance and modelling uncertainties due to plant wear over its operating life. Quantitative definitions are made in the time domain [1, 45] and in the frequency domain [43]

One definition of robustness in the complex frequency (s) domain is the complementary sensitivity function, $T(s)$, of (2.6). It is reasonable to suppose that as the sensitivity reduces, the robustness increases, and this is certainly the case according to (2.7). In the frequency domain, this may be a Bode magnitude plot, $T_{\mathrm{dB}}(\omega) = 20\log_{10}\left|T(j\omega)\right|$. Then a lower threshold boundary, $T_{\mathrm{dB\,min}}(\omega)$, is specified for $T_{\mathrm{dB}}(\omega)$. The specification is satisfied if $T_{\mathrm{dB}}(\omega) > T_{\mathrm{dB\,min}}(\omega)$ over the frequency range that the control system is required to operate. Usually, $0 \le \omega \le \omega_{\mathrm{b}}$, where ω_{b} is the control system bandwidth defined as the lowest angular frequency at which $T_{\mathrm{dB}}(\omega)$ falls below $T_{\mathrm{dB}}(0)$ by $3[\mathrm{dB}]$.

Maintenance of robustness in the frequency domain, however, does not imply that a given transient response specification is satisfied. One way of assessing this requirement is to examine variations of the step response with respect to the specified one that result from the plant parameters being changed. The robustness can then be defined as *how little* the step response changes with respect to the specified step response for given plant parameter changes.

Care must be taken to specify a sufficient number of parameters of the step response. For example, specifying the settling time alone according to the 5% criterion would be insufficient because of the many different shaped step responses that are possible which will have the same settling time. Some of these responses would be unacceptable due to insufficient damping, as illustrated by the step response sketched in Figure 2.2.

Specifying the percentage overshoot too would be sufficient in most cases. If a non-oscillatory step response is needed, the percentage overshoot would be specified as zero.

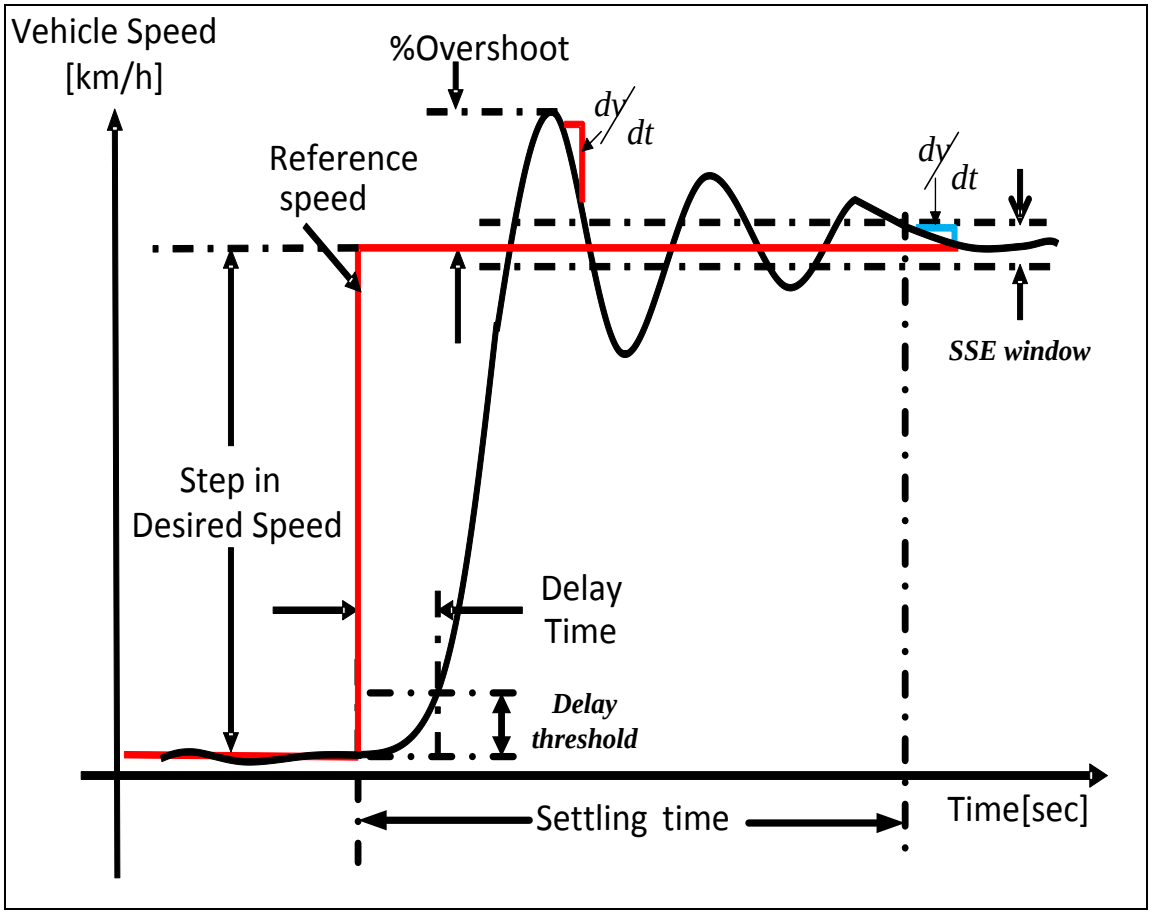


Figure 2-3: Step response specification parameters

The delay time illustrated is due to dynamic lag and its proportion relative to the settling time depends upon the order of the closed loop system. In some applications, however, such an effect could be largely due to pure time delay in the plant, sometimes referred to as transport delay. This is a property of some plants that cannot be modified by feedback control, and the step response of the closed loop system has to include this time delay. Under these circumstances, the settling time has to be chosen larger than the transport delay for the specification to be realisable.

Note: $\frac{dv}{dt}$ is the maximum allowable rate of change during control and stabilization.

2.4 Internal Stability Definition

If a closed loop system is internally stable, it contains no unstable modes. The reason for introducing this topic is that it is possible to find closed loop systems that *appear* to be stable through having closed loop transfer functions with poles in the left half of the *s*-plane, but are, in fact, unstable due to the cancellation of zeros in the right half of the *s*-plane by closed loop poles in the same locations. In theory this could occur in a controller designed by pole assignment but the control system designer would be aware of the right half plane (RHP) zeros and would never attempt to cancel them. On the other hand, the RHP zeros may be overlooked when using a sliding mode controller with a boundary layer, the design of which only requires knowledge of the plant relative degree [64, 65]. A similar issue could potentially occur with the observer based robust control (OBRC) of Chapter 4. As the width of the boundary layer is reduced in the sliding mode control, then this is equivalent to increasing the proportional gain in a unit feedback control loop, so root loci of the system closely approach any RHP zeros, causing instability. The unstable mode caused is *internal* in the sense that it cannot be detected by observing the output response to the reference input. The system could then be described as *internally unstable*.

A closed loop system that is *internally stable* is one whose characteristic equation has roots with negative real parts. This is the characteristic equation of the system matrix of the state space model of the closed loop system. Thus internal stability ensures that there are no unstable modes of the closed loop system, whether or not they can be detected by observing the output response to the reference input. It follows from the above that a feedback control system is internally stable if it appears to be stable by observing the output and reference input and there is no RHP pole zero cancellation.

Returning to the unit feedback control system of section 2.1,its components are defined as $T(s), S(s), L(s) = P(s)C(s)$ and $\mathrm{C(s)S(s)}$. Fig. 2.3 shows the relevant control system block diagram, which includes external disturbance inputs, $d_i(s)$ and $d_o(s)$, referred to both the input and output of the plant.

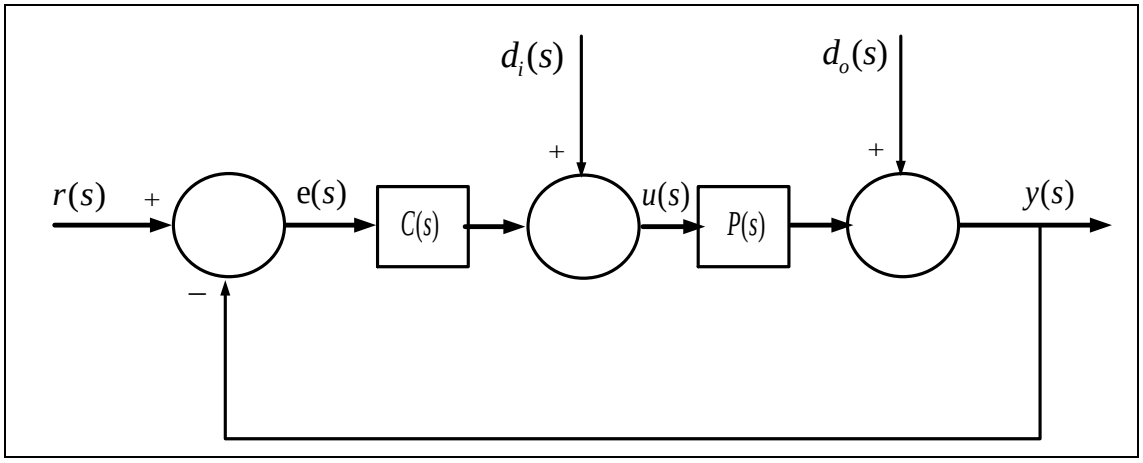


Figure 2-4: SISO Linear System Block Diagram for Internal Stability Analysis

The transfer function relationships of this diagram are as follows:

$$u(s)_{|r(s)=0} = \frac{1}{1+C(s)P(s)} d_i(s) + \frac{C(s)}{1+C(s)P(s)} d_o(s) \tag{2.8}$$

$$y(s)_{|r(s)=0} = \frac{P(s)}{1+C(s)P(s)} d_i(s) + \frac{1}{1+C(s)P(s)} d_o(s) \tag{2.9}$$

In view of (2.6), however, the sensitivity and complementary sensitivity are given, respectively, by

$$S(s) \equiv \frac{1}{(1+C(s)P(s))} \tag{2.10}$$

and

$$T(s) \equiv \frac{C(s)P(s)}{(1+C(s)P(s))} \tag{2.11}$$

Therefore (2.8) and (2.9) may be written as

$$u(s) = S(s)d_i(s) + C(s)S(s)d_o(s) \tag{2.12}$$

$$y(s) = P(s)S(s)d_i(s) + S(s)d_o(s) \tag{2.13}$$

Therefore, it can be said the system is internally stable if the transfer functions, $S(s)$, $C(s)S(s)$ and $P(s)S(s)$ are each asymptotically stable.

By using an example, the meaning of an internally stable system can be shown clearly. Consider a plant with transfer function given by

$$P(s) = \frac{1}{(s-1)(s+2)} \tag{2.14}$$

and its controller given by

$$C(s) = \frac{(s+a)}{(s+b)} \tag{2.15}$$

The unity feedback structure of Figure 2-3 is assumed. Therefore, the open loop transfer function is given by

$$L(s) = P(s)C(s) = \frac{(s+a)}{(s-1)(s+2)(s+b)} \tag{2.16}$$

Clearly, the RHP pole-zero cancellation is avoided if $a \neq -1$ in the controller design. There is no cancellation of the RHP zeros or poles and therefore, the RHP pole remains in $L(s)$. The sensitivity transfer function is given by equation 2.10 and can be written for the closed loop of the plant. Thus

$$S(s) = \frac{1}{1+C(s)P(s)} = \frac{1}{1+\frac{(s+a)}{(s+b)}.\frac{1}{(s-1)(s+2)}} = \frac{(s+b)(s-1)(s+2)}{(s+b)(s-1)(s+2)+(s+a)} \tag{2.17}$$

It can be seen that there is no pole zero cancellation if $a \neq -1$.Next,

$$C(s)S(s) = \frac{(s+a)}{(s+b)}.\frac{(s+b)(s-1)(s+2)}{(s+b)(s-1)(s+2)+(s+a)} = \frac{(s+a)(s-1)(s+2)}{(s+b)(s-1)(s+2)+(s+a)} \tag{2.18}$$

Again, there is no pole-zero cancellation if $a \neq -1$. Finally

$$P(s)S(s) = \frac{1}{(s-1)(s+2)}.\frac{(s+b)(s-1)(s+2)}{(s+b)(s-1)(s+2)+(s+a)}$$

$$= \frac{(s+b)}{(s+b)(s-1)(s+2)+(s+a)} \tag{2.19}$$

This component would be unstable with $a = -1$ because $(s-1)$, would become a factor of the denominator. The conclusion of this analysis is that the control system designer

would be free to try and find values of c and $a \neq -1$ yielding internal stability by forcing all three closed loop poles to be in the left half of the s-plane. It is important to realise, however, that as it stands, the controller only has two adjustable parameters, c and a, and therefore design by pole assignment, in which all three closed loop poles can be chosen at the outset to achieve a specified transient response, cannot be done. This problem could be solved, however, by introducing a proportional gain, K, in the controller, so that $C(s) = K\dfrac{(s+a)}{(s+b)}$.

2.5 Control System Design Methodology

Assuming that the plant, the controller and therefore the closed loop system are linear, there are three basic control system design methodologies, as follows,

I: - Pole assignment to achieve a specified closed loop transient response. The settling time formula of section 2.6 renders this methodology straightforward for a linear plant of arbitrary order.

II: - Optimal control [51, 52, 63, 42], meaning the use of an optimisation technique to determine the controller gains that minimise (or maximise) a given cost function while respecting a set of constraints.

III: - Determination of the controller gains basically by tuning in real time on the real plant but often proceeded by tuning using a simulation [46, 45].

Methodologies I and II are model based Methodology I is carried out in the time domain but methodology II may be carried out in the time domain or the frequency domain.

The IMC, OBRC and the RFH controllers are compatible with design methodology I for plants of any order. This is also true of the linear state feedback controller [1, 48,59]. The standard PID controller and its variants, i.e., the PI, PD, P (proportional) and similar controllers of slightly different structure, are also compatible with design methodology I but only for plants of limited order (second order for the PID and PD controllers and first order for the PI and P controllers).

The H-infinity design approach applies to the design of a control system with the classical unit feedback structure and is an example of methodology II in the frequency domain [9, 23, 32, 40, 41, 43, and 50]. Formulating constraints for the H-infinity method is an important part of the design process, since the optimisation tools will exploit the weakness in ill-defined constraints and consequently the solution will not be a robust one. Also, the H-infinity approach is most effective in assuring closed loop stability in cases of extreme plant model uncertainty. The order of the resulting controller, $C(s)$, depends on the plant transfer function and the optimisation criterion. The RST controller has a different structure and is another example of methodology II in the frequency domain [26, 35, and 36]. Both of these require an optimisation tool in order to design the controller.

The linear state feedback controller is also compatible with design methodology II in the time domain when the gains are determined by the LQ method. Here, the plant state space model and the weighting parameters of a linear quadratic integral cost function are used in a matrix Ricatti equation to calculate the state feedback gains that minimise the integral cost function [27, 66].

The standard PID controller and its variants are the only controllers lending themselves to design methodology III but a plant model is not required. The tuning is essentially on the basis of experience with specific types of plant but, with a new plant, has initially to be by trial and error to build up the experience. It is important to note that this process increases considerably in difficulty as the plant order increases beyond three and it may be impossible in some cases to achieve a specified performance, in which case the more sophisticated control techniques referred to above have to be considered.

Integral anti-windup is strongly recommended for any controller containing integral terms to minimised issues due to control saturation. Also, and software differentiation with inbuilt Measurement noise filtering is recommended in any controller requiring a differential term to avoid momentary control saturation.

2.6 The Settling Time Formula

The Dodds settling time formula [46] is a means of control system design by pole assignment to achieve a specified settling time with zero overshoot for a linear control system of arbitrary order. For working with this formula, the settling time is defined as the time taken to *nominally* reach the steady state value from one reference input level to another either from a step up or a step down, noting that the transient response of any continuous linear system takes an infinite time to reach a constant steady state value. According to the 5% criterion, for a closed loop system with a unity closed loop DC gain, it is the time taken for the magnitude of the error, $e(t) = r(t) - y(t)$ to fall to and thereafter remain below 5% of its peak value. This time is measured from the instant at which the peak occurs. If the poles of the closed loop system are made approximately coincident at $s_{1,2,\ldots,n} = -1/T_c$, where T_c is the time constant of the n identical first order subsystems connected in a chain to give the desired closed loop transfer function,

$$\frac{Y(s)}{R(s)} = \left(\frac{1}{1+sT_c}\right)^n \tag{2.20}$$

The settling time formula provides a fairly accurate response up to a 6^{th} order system and can be easily corrected in one step to be applicable to systems of higher order than this. The 5% formula is as follows:

$$T_s = 1.5(1+n)T_c \tag{2.21}$$

where n is the order of the closed loop system. In view of equation (2.19), the desired closed loop pole locations can be expressed in terms of T_s and n as follows:

$$s_{1,2,\ldots,n} = -1/T_c = 1.5(n+1)/T_s \tag{2.22}$$

Hence, the general characteristic equation for the pole placement is given by

$$\left[s + \frac{1.5(n+1)}{T_s}\right]^n = 0 \tag{2.23}$$

3 Observer Based Engine Load Estimation

3.1 Overview

The aim is to estimate an external disturbance applied to a plant, with a view to enhancing its control. Today's production does not have a sensor to measure the torque at the engine flywheel. Hence, the torque signal used in the system is estimated by calibrating a series of lookup tables using empirical methods. This is done under the steady state condition, $Torque = f(fuel, Enginespeed, ...)$. The alternative method presented in this thesis is novel with respect to the field of road vehicle control. The uniqueness of the method will be explained in this chapter.

In general, an observer [5, 49, 28] is designed to estimate the state of a plant in cases where the state variables are required for use in a state feedback control law but cannot be measured. This uses a plant model that matches the real plant as closely as possible. The term, 'state observer' is used frequently in the literature but is really misleading as it is not the state that is observed, since it is not available. Otherwise, the observer would not be needed. It is actually the available signals that are observed, i.e., the plant control variable, $u(t)$, and the measurement variable, $y(t)$. The term, 'observer,' comes from the property of *observability* that a plant must have for it to be possible to estimate the state, $\mathbf{x}(t)$. If a plant is observable, then, with the aid of an accurate state space model of the plant, it is possible to estimate the present plant state, $\mathbf{x}(t)$, using past observations of $u(\tau)$ and $y(\tau)$, $0 \le \tau < t$. In view of the above, the author prefers the term "estimator," to, "observer2," but the terms may be used interchangeably. In fact, the term "state estimator," is also used instead of, "observer," in the literature.

In the field of electric drives, an observer is often extended to be able to estimate an external disturbance, $d(t)$, referred to the control input, as well as the plant state [48]. Here, this idea is extended further to enable an observer to work with a greatly simplified plant model that is not necessarily well matched to the real plant. It is this type of observer that is employed in the OBRC fully described in Chapter 4. Such an observer will be developed for application to a heavy duty vehicle later on in this,

chapter. In preparation for this, the vehicle, model hitherto referred to as the plant model, will be fully discussed.

The plant is composed of several components comprising the engine, gearbox, drive shaft, differential and the vehicle mass. There is also a considerable external disturbance consisting of the gravitational force acting on the vehicle due to its inclination about the pitch axis. This can be regarded as
a relatively complex system [26, 25]. The simplified model to be used in the estimator, however, does not have to contain details such as the different gear ratios that may be selected, the mass of the payload or the gravitational load force. This is because the combined effect of all of these is equivalent to an external load torque applied to the flywheel of the engine. An estimate of this equivalent external load torque and its counteraction by a controller will be sufficient to handle all the aforementioned physical influences. Hence, the simplified model consists just of the engine alone (disengaged from the drive line).

3.2 The Inverse Dynamic Load Representation

The dynamics of a mechanical system model is defined as the part that yields rotational and/or translational velocities when forces and/or torques are applied. The definition of the inverse dynamics then follows naturally as a reformulation of the dynamics as the part of the model that yields the forces and/or torques needed to produce given translational and/or rotational velocities. The plant model will be divided into the following two parts:

1) The dynamics model of the unloaded diesel engine
2) The inverse dynamics model of the remainder of the vehicle

A typical unloaded diesel engine can be represented by the transfer function of a first order lag. Hence part (1) of the plant model is:

$$G_e(s) = \frac{y_v(s)}{u(s)} = \frac{b}{(s+a)} \tag{3.1}$$

where $u(s)$ is the throttle input, i.e., the fuel volume flow rate, but this is scaled to be numerically equal to the torque developed by the engine. $y_v(s)$ is the measurement of the flywheel speed. The constant parameters, a and b, are determined from identification tests on the engine.

Figure 3-1 shows two models of the vehicle in block diagram form.

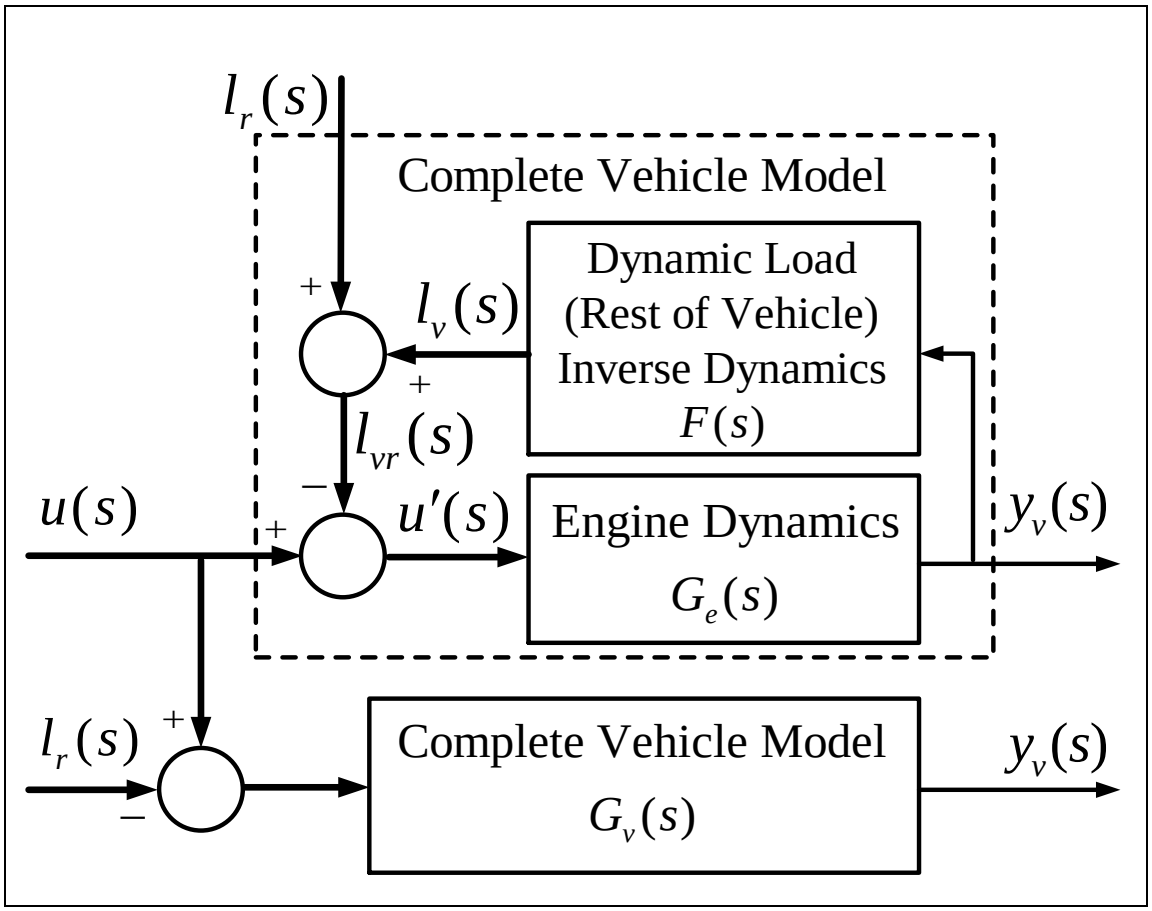


Figure 3-1: Mathematical representation of the inverse dynamics

$G_v(s)$ is the overall transfer function and is a basic form of model that is directly derived from experimental data. $l_r(s)$ is the external load torque acting is the overall transfer function, while the one at the top is decomposed into parts (1) and (2) specified above and connected according to the inverse dynamics principle.

The upper two blocks constitute an alternative model decomposed into parts (1) and (2) specified above and connected according to the inverse dynamics principle. Here, Figure 3-2 represents these two models pictorially.

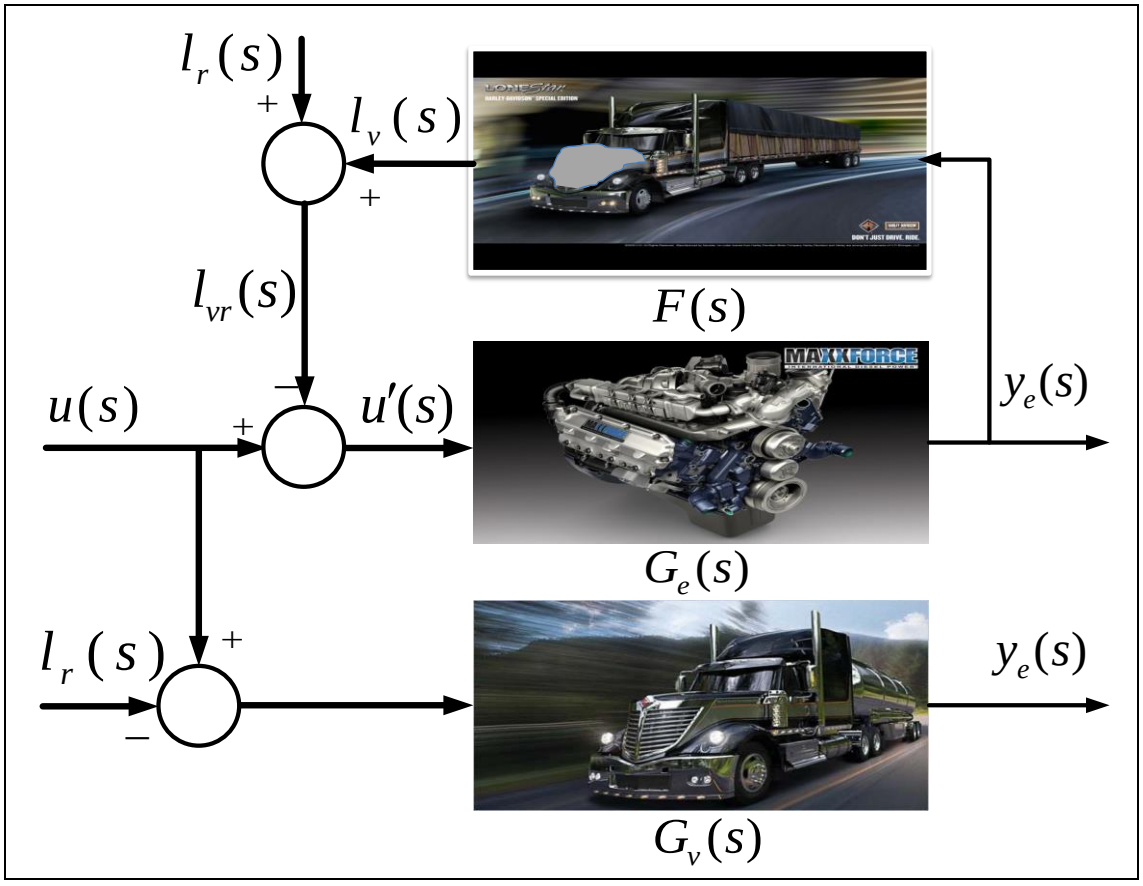


Figure 3-2: Pictorial representation of the inverse dynamics

In Figures 3-1 and 3-2, $u(s)$ and $y_v(s)$ are physical signals while the dynamic load torque, $l_v(s)$, the external load torque, $l_r(s)$ the net load torque, $l_{vr}(s)$, and the fictitious net engine input, $u'(s)$,do not physically exist but are variables of the mathematical model. So the vehicle is actually represented as the engine subjected to a l load torque represented mathematically by a signal, $l_{vr}(s)$, which is acting at the same point as the control input, $u(s)$.

3.3 Transfer Function for *F(s)*

To determine $F(s)$ from $G_v(s)$ and $G_e(s)$ the equivalence of the upper and lower block diagrams of Figure 3-1 required, meaning that the transfer functions must be the same. For this purpose, the external inputs are not needed as the model is a linear one. Hence Figure 3-3 shows the basic inverse dynamic block diagram, which will be used to determine the $F(s)$.

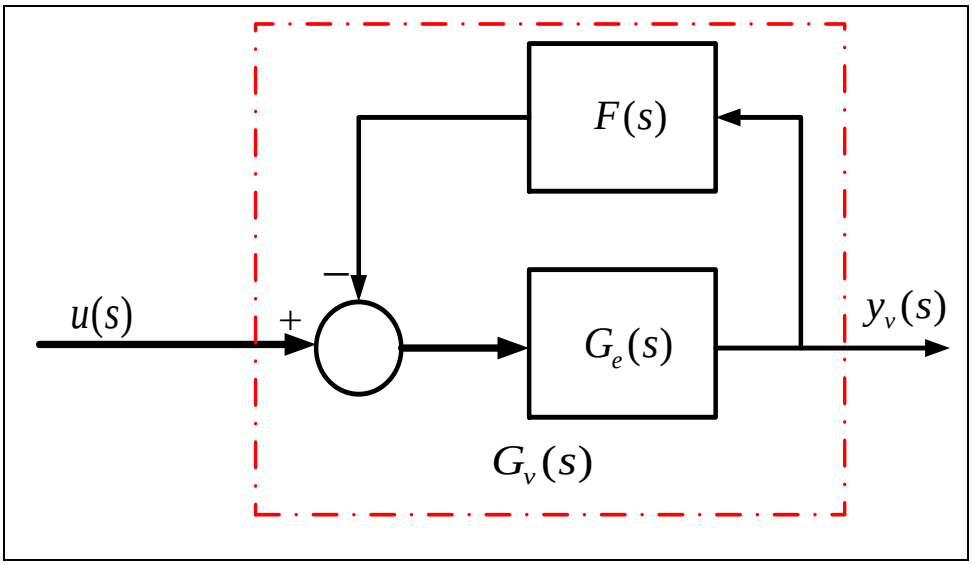


Figure 3-3: Simplified block diagram of the vehicle with the dynamic engine load in the inverse dynamic form

The transfer function of this model is given by

$$G_v(s) = \frac{y_v(s)}{u(s)} = \frac{G_e(s)}{1 + F(s)G_e(s)} \Rightarrow F(s) = \frac{1}{G_e}\left[\frac{G_e(s)}{G_v(s)} - 1\right] \tag{3.2}$$

where

$$G_v(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{3.3}$$

It should be noted that identification tests on vehicles yield such fourth order models with two finite zeros. Substituting for $G_v(s)$ and $G_e(s)$ in equation (3.1) using equations (3.2) and (3.1) then yields

$$\begin{aligned} F(s) &= \frac{(s+a)}{b}\left[\frac{s^3+a_2s^2+a_1s+a_0}{b_2s^2+b_1s+b_0}\frac{b}{(s+a)}-1\right] \\ &= \frac{s^3+a_2s^2+a_1s+a_0}{b_2s^2+b_1s+b_0}-\frac{(s+a)}{b} \\ &= \frac{b(s^3+a_2s^2+a_1s+a_0)-\left[b_2s^3+(b_2a+b_1)s^2+(b_1a+b_0)s+b_0a\right]}{b(b_2s^2+b_1s+b_0)} \\ &= \frac{(b-b_2)s^3+(ba_2-b_2a-b_1)s^2+(ba_1-b_1a-b_0)s+(ba_0-b_0a)}{b(b_2s^2+b_1s+b_0)} \end{aligned} \tag{3.4}$$

Since this transfer function has negative relative degree, ie, degree of denominator minus degree of numerator, then it cannot be implemented directly in a Matlab/Simulink diagram. There two possible approaches to overcome this issue.

1. Long division to obtain the sum of the quotient that is a pure derivative and the remainder that is a transfer function with zero relative degree, both of which can be modelled using Matlab/Simulink.

2. Treat the block diagram shown in Figure 3.1 as a black box and derive the transfer function relationships between the inputs $u(s)$ and $l_r(s)$ and the outputs $y_v(s), l_v(s)$ and $u'(s)$. The component transfer functions then all have non-negative relative degree and can therefore be modelled using Matlab/Simulink.

Option 1 is chosen as it appears to be the simpler one.

Let transfer function (3.4) be simplified to the standard form before doing the long division. Thus

$$F(s) = \frac{1}{b}\left[\frac{c_3 s^3 + c_2 s^2 + c_1 s + c_0}{b_2 s^2 + b_1 s + b_0}\right] \tag{3.5}$$

where $c_3 = b - b_2$, $c_2 = ba_2 - b_2 a - b_1$, $c_1 = ba_1 - b_1 a - b_0$ and $c_0 = ba_0 - b_0 a$. The long division working is as follows.

$$\begin{array}{r} \frac{c_3}{b_2}s \\ b_2 s^2 + b_1 s + b_0 \overline{)\, c_3 s^3 + c_2 s^2 + c_1 s + c_0} \\ \underline{c_3 s^3 + \frac{c_3 b_1}{b_2} s^2 + \frac{c_3 b_0}{b_2} s} \\ \underbrace{(c_2 - \frac{c_3 b_1}{b_2})}_{d_2} s^2 + \underbrace{(c_1 - \frac{c_3 b_0}{b_2})}_{d_1} s + c_0 \end{array}$$

Hence

$$F(s) = \frac{1}{b}\left[\frac{c_3}{b_2} s + \frac{d_2 s^2 + d_1 s + c_0}{b_2 s^2 + b_1 s + b_0}\right] \tag{3.6}$$

Now a passenger car will be taken as a numerical example. The overall transfer function and engine transfer function are derived from experimental data obtained from a 2.0 litre diesel vehicle using PRBS “Pseudo Random Binary Sequence”, in conjunction with the Matlab system identification toolbox, as follows:

$$G_v(s) = \frac{475\text{s}^2 + 285.6\text{s} + 9236}{\text{s}^3 + 7.596\text{s}^2 + 254.4\text{s} - 0.2454} \tag{3.7}$$

$$G_e(s) = \frac{176.24}{\text{s} - 0.04668} \tag{3.8}$$

Hence the long division yields

$$\text{F(s)} = -0.0036\text{s} + \frac{0.01499\text{s}^2 + 0.4759\text{s} + 0.004633}{\text{s}^2 + 0.6013\text{s} + 19.44} \tag{3.9}$$

It is important to note that Diesel engines are open loop unstable, as indicated by transfer function (3.8). The negative constant term in transfer function (3.7) also indicates open loop instability of the complete vehicle. Hence closed loop control is essential.

3.4 The Concept of Observer Based Load Estimation

Figure 3-4 represents a first order plant with control input, $u(s)$, disturbance input, $d(s)$, and output, $y(s)$ together with a model with the same input, additional input, $l(s)$, and output, $\hat{y}(s)$.

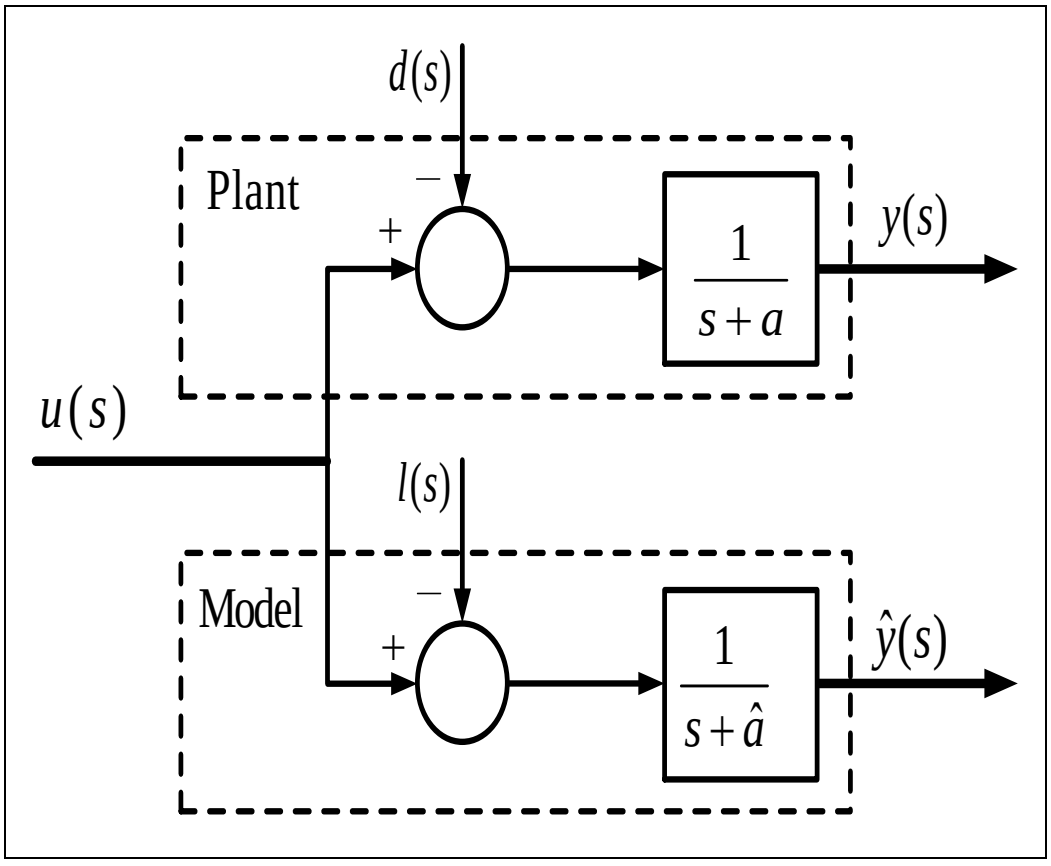


Figure 3-4: A first order plant and its model

It will now be shown that there exists an additional input, $l(s)$, that yields $\hat{y}(s) = y(s)$. This is not only the basis of the observer based load estimation but also that of the OBRC of Chapter 4. The transfer function relationships of Figure 3-4 are as follows:

$$y(s) = \frac{1}{s+a}\left[u(s) - d(s)\right] \tag{3.10}$$

and

$$\hat{y}(s) = \frac{1}{s+\hat{a}}\left[u(s) - l(s)\right] \tag{3.11}$$

If $\hat{y}(s) = y(s)$, then the RHS of equations equation (3.10) and (3.11) must be equal. Therefore

$$\frac{1}{s+\hat{a}}\left[u(s) - l(s)\right] = \frac{1}{s+a}\left[u(s) - d(s)\right] \Rightarrow$$

$$u(s) - l(s) = \frac{s+\hat{a}}{s+a}\left[u(s) - d(s)\right] \tag{3.12}$$

Hence rearranging equation (3.13) to make $l(s)$ the subject yields

$$l(s) = \frac{a-\hat{a}}{s+a}u(s) + \frac{s+\hat{a}}{s+a}d(s) \tag{3.13}$$

The first term on the right hand side of equation (3.12) is the component of $l(s)$ needed to compensate for the plant model parametric error, $a - \hat{a}$, while the second term is the component of $l(s)$ that compensates for the external disturbance, $d(s)$. It can be seen that if the plant and model parameters are the same, i.e., $a = \hat{a}$, then $l(s) = d(s)$. On the other hand, if the plant model parameter is mismatched and $d(s) = 0$, then $l(s)$ can be used as a plant model parameter correction under steady state conditions. Thus, if $u(t) = const. = u_0$, then $u(s) = u_0/s$ and then equation (3.13) yields the following steady state value of $l(t)$:

$$l_{ss} = \lim_{s \to 0} s.\frac{a-\hat{a}}{s+a}.\frac{u_0}{s} = \left(1 - \frac{\hat{a}}{a}\right)u_0 \tag{3.14}$$

from which the true value of the plant parameter is

$$a = \frac{u_0\hat{a}}{u_0 - l_{ss}} \tag{3.15}$$

The key to the usefulness of this is the special type of estimator that could, in this example, be used to estimate $l(t)$ and therefore l_{ss}. Then the parameter estimate, $\hat{a}$, would be updated by replacement with the more accurate estimate given by equation (3.15).

Figure 3-5 shows the overall structure of the system providing an estimate, $\hat{l}_{vr}(s)$, of the net engine load torque, $l_{vr}(s)$, shown in Figure 3-1.

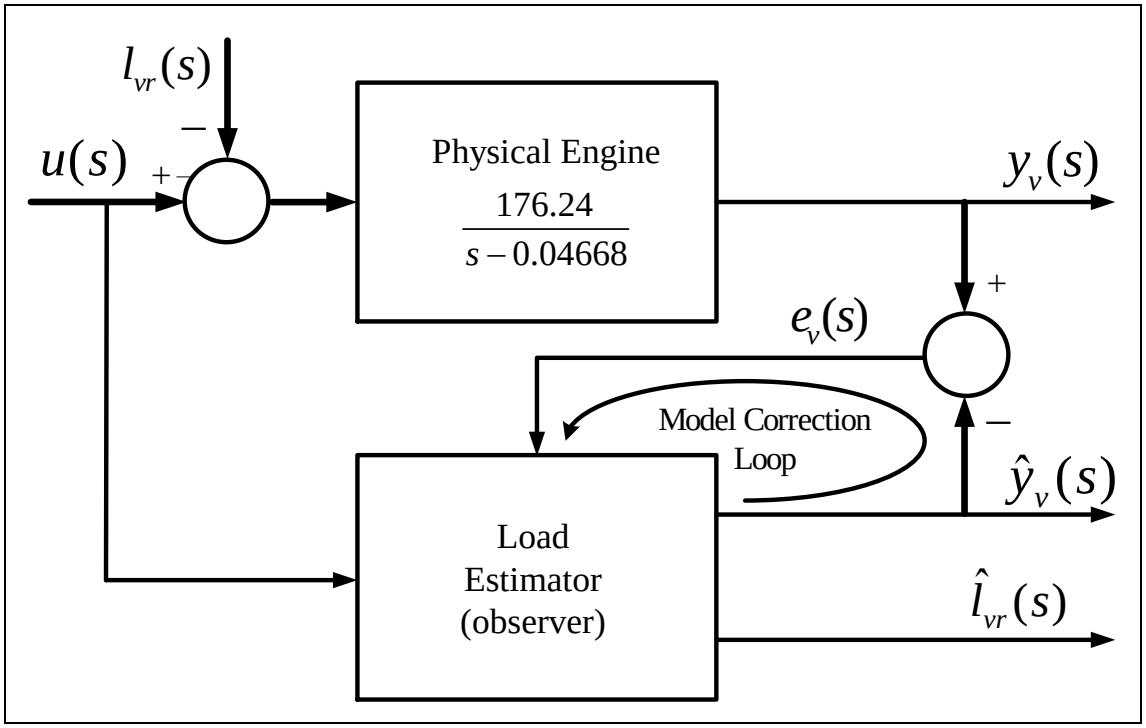


Figure 3-5: Block diagram of the external load estimator

Essentially, the load estimator contains a model of the engine driven by the same input, $u(s)$, as applied to the physical engine. This generates an estimate, $\hat{y}_v(s)$ of the measured engine speed, $y_v(s)$, together with the required engine load torque estimate, $\hat{l}_{vr}(s)$. An error, $e_v(s) = y_v(s) - \hat{y}_v(s)$, is formed and fed back to the engine model to correct its output so that $e_v(t) \to 0$. If the engine model is accurate, then once $e_v(t) \cong 0$, $\hat{l}_{vr}(t)$ is a good estimate of $l_{vr}(t)$.

The following section presents the observer and the steps of its design.

3.5 The Observer and its Design

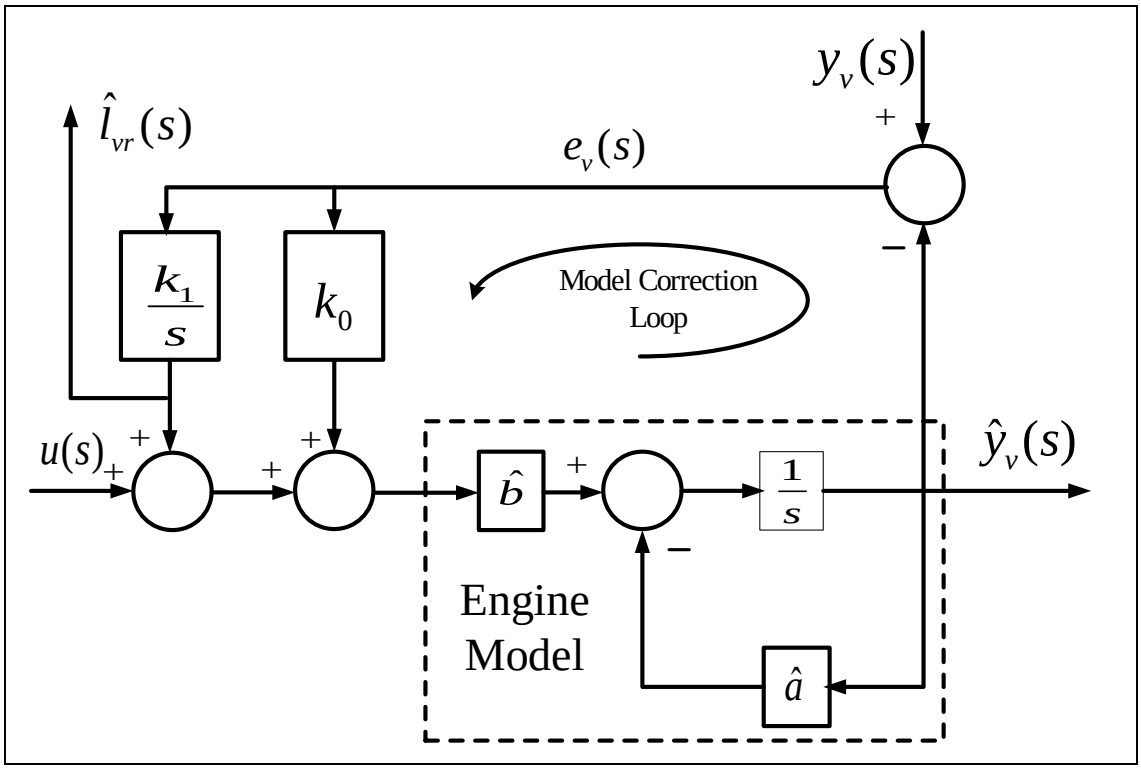


Figure 3-6: Observer block diagram for diesel engine

The only purpose of this observer is to estimate the engine torque, as there is only one model state variable, $\hat{y}_v(s)$, and the real value, $y_v(s)$, is already available as a measurement. The special feature enabling estimation of $l_{\text{vr}}(s)$ is the integrator with the gain, k_1. The key to this is regarding $l_{\text{vr}}(t)$ as if it was a state variable and supposing that it it may be regarded constant. As will be seen, the observer still works with it time varying. On this assumption, the state differential equation for $l_{\text{vr}}(t)$ is

$$\dot{l}_{\text{vr}}(t) = 0 \Rightarrow l_{\text{vr}} = const. \tag{3.16}$$

The corresponding observer state differential equation is arranged as follows:

$$\dot{\hat{l}}_{\text{vr}}(t) = k_1 e_v \tag{3.17}$$

where k_1 is a constant gain. This enables e_v to continually correct $\hat{l}_{\text{vr}}$ until the correction loop drives e_v to approximately zero, whereupon $\hat{l}_{\text{vr}}$ will have converged almost to the correct value. From equation (3.17),

$$\hat{l}_{\text{vr}}(t) = k_1 \int_0^t e_v(\tau) d\tau \Rightarrow \hat{l}_{\text{vr}}(s) = \frac{k_1}{s} e_v(s) \tag{3.18}$$

The observer block diagram agrees with this.

To calculate the observer gains, k_0 and k_1, in the block diagram of Figure 3.6, the method of pole assignment is used. the first the characteristic equation of the observer is needed in terms of these gains. This is obtained in straightforward fashion by equating the determinant of Mason's Rule to zero. The signal flow graph is not needed as the block diagram contains the same information Hence

$$1 - \left[-\frac{1}{s}\left(\hat{a} + \hat{b}k_0 + \frac{\hat{b}k_1}{s} \right) \right] = 0 \Rightarrow s^2 + \left(\hat{a} + \hat{b}k_0 \right) s + \hat{b}k_1 = 0 \tag{3.19}$$

It should be noted, however, that in cases of higher order, particularly those having a complex loop structure, it will be more economical in time and effort to use the numerical method of Appendix A2.

The Dodds 5% settling time formula (2.11will now be used for the pole placement design to achieve a correction loop settling time of T_{so}. Thus

$$T_{so} = 1.5(1+n)T_{co} \tag{3.20}$$

By inspection of Figure 3.6 the order of the observer is $n = 2$ and the two poles are placed at

$$s_{1,2} = -\frac{1}{T_{co}} = \frac{-1.5(1+n)}{T_{so}}\Big|_{n=2} = -\frac{9}{2T_{so}} \tag{3.21}$$

Hence, the desired characteristic polynomial is given by

$$(s + \frac{9}{2T_{so}})^2 = s^2 + \frac{9}{2T_{so}} s + \frac{81}{4T_{so}^2} \tag{3.22}$$

This must be the same as the characteristic polynomial of equation (3.19). Thus

$$s^2 + (\hat{a} + \hat{b}k_0)s + \hat{b}k_1 = s^2 + \frac{9}{2T_{so}}s + \frac{81}{4T_{so}^2}$$

Equating the coefficients of like powers of s then yields the following formulae for the observer gains:

$$k_0 = \frac{1}{\hat{b}}\left(\frac{9}{T_{so}} - \hat{a}\right) \quad \text{and} \quad k_1 = \frac{81}{4T_{so}^2\hat{b}} \tag{3.23}$$

The observer variable, $\hat{y}_v(t)$, will track the measured signal, $y_v(t)$ and $\hat{l}_{\text{vr}}(t)$ will track the time varying engine load torque, $l_{\text{vr}}(t)$ more closely if the observer settling time is made smaller. Typically the $T_{so} < 5T_s$, where T_s is the settling time of the control system of which the observer is part. Transients occur due to load changes, which may come from the driving conditions or gear changes. Hence, the value for T_{so} is chosen to be considerably smaller than the typical settling time of the engine speed. The vehicle is much more responsive to the driver demand when the vehicle is unloaded than when it is fully loaded. A typical 10litre engine will accelerate from low idle (500 rpm) to high idle (2000 rpm) in approximately 1.5 sec. Hence a value of $T_{so} = 0.01\text{sec}$ was chosen, yielding $k_1 = 1149$ and $k_0 = 5.109$.

3.6 Passenger Car Inverse Dynamic Simulation

3.6.1 Overview

In this section, the simulation results of the inverse dynamic model of the vehicle are validated by comparison with those of the standard form of the vehicle model, which should be the same. The load torque observer is included in both simulations to check its performance. The simulation is performed in two modes: open and closed loop.

3.6.2 Observer Simulation Results

In the open loop mode, both plant models and the observer are driven by a piecewise constant common control torque, $u(t)$, as shown in Figure 3-5

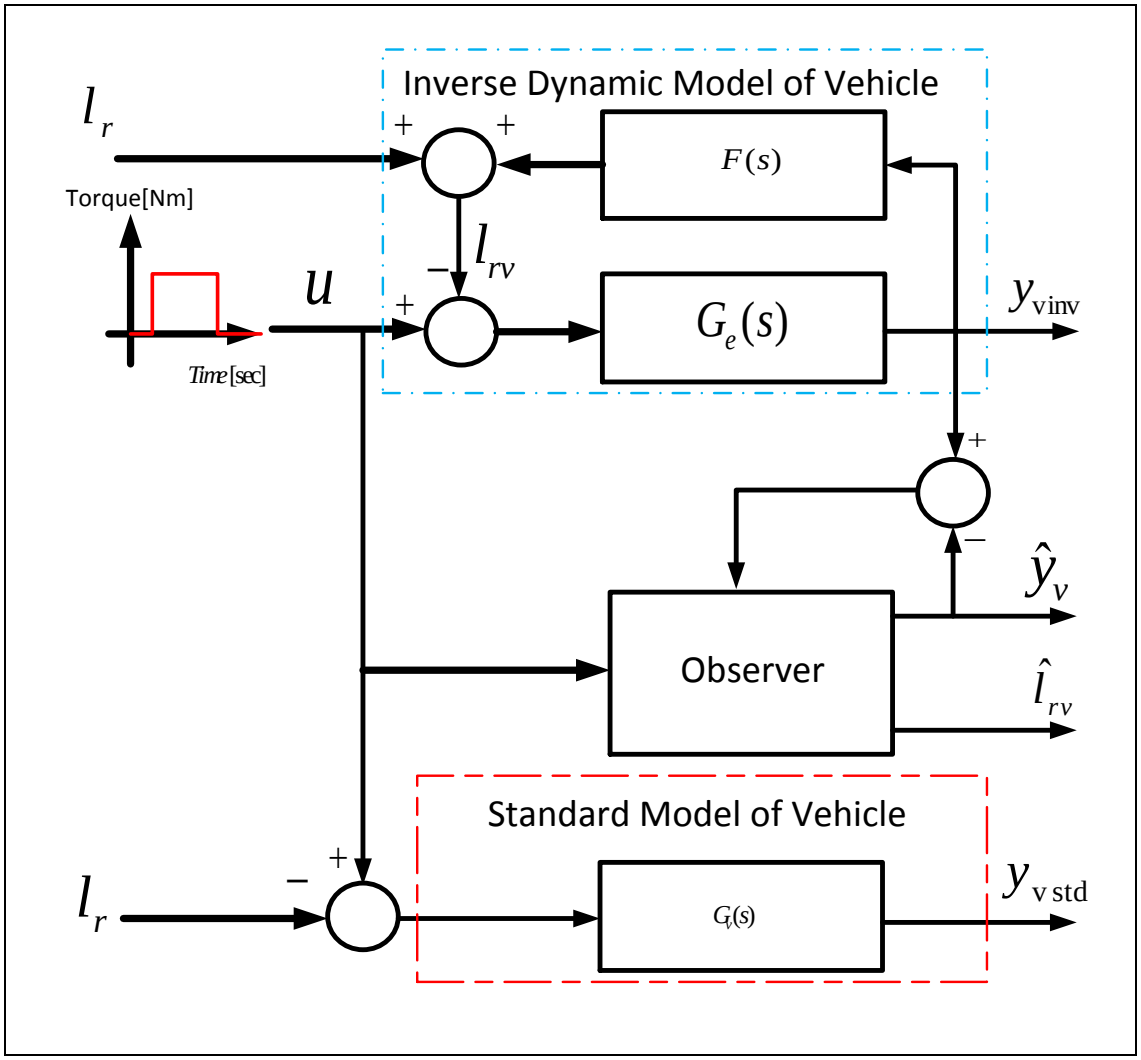


Figure 3-5 Open loop block diagram of inverse dynamics and vehicle model with observer

The open loop input, $u(t)$, is a step from zero to a constant positive value at $t = 1[s]$ followed by another step returning to zero at $t = 8[s]$.

Figure 3.6 shows the plot of $y_v(t)$, $\hat{y}_v(t)$ and the observer error, $e_{vm}(t) = y_{vinv}(t) - y_{vstd}(t)$.

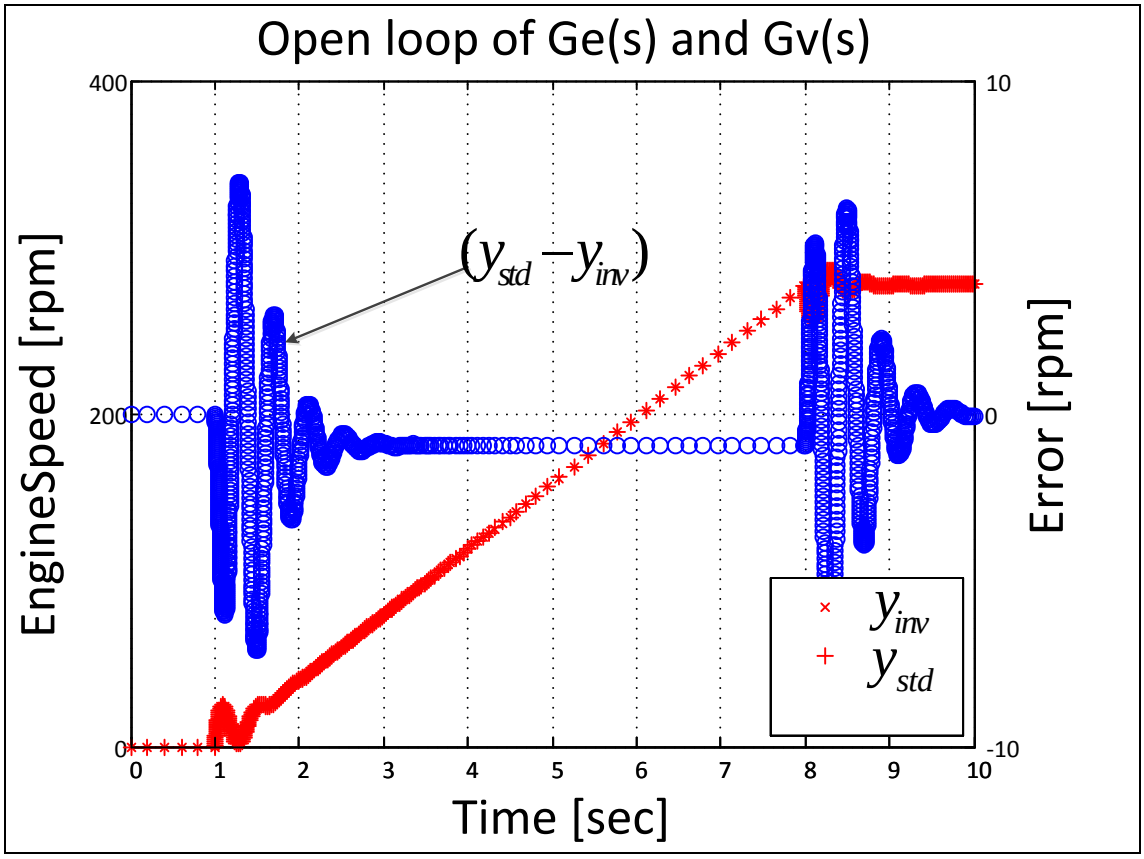


Figure 3-6 Open loop speed response of inverse dynamic and standard models

The oscillation at the start and end of the step is due to the complex conjugate poles of $G_v(s)$. The error $e_v(t) = y_{std} - y_{inv}$ is small as shown Figure 3.6.

Figure 3.7 shows the ability of the observer to estimate the external load torque.

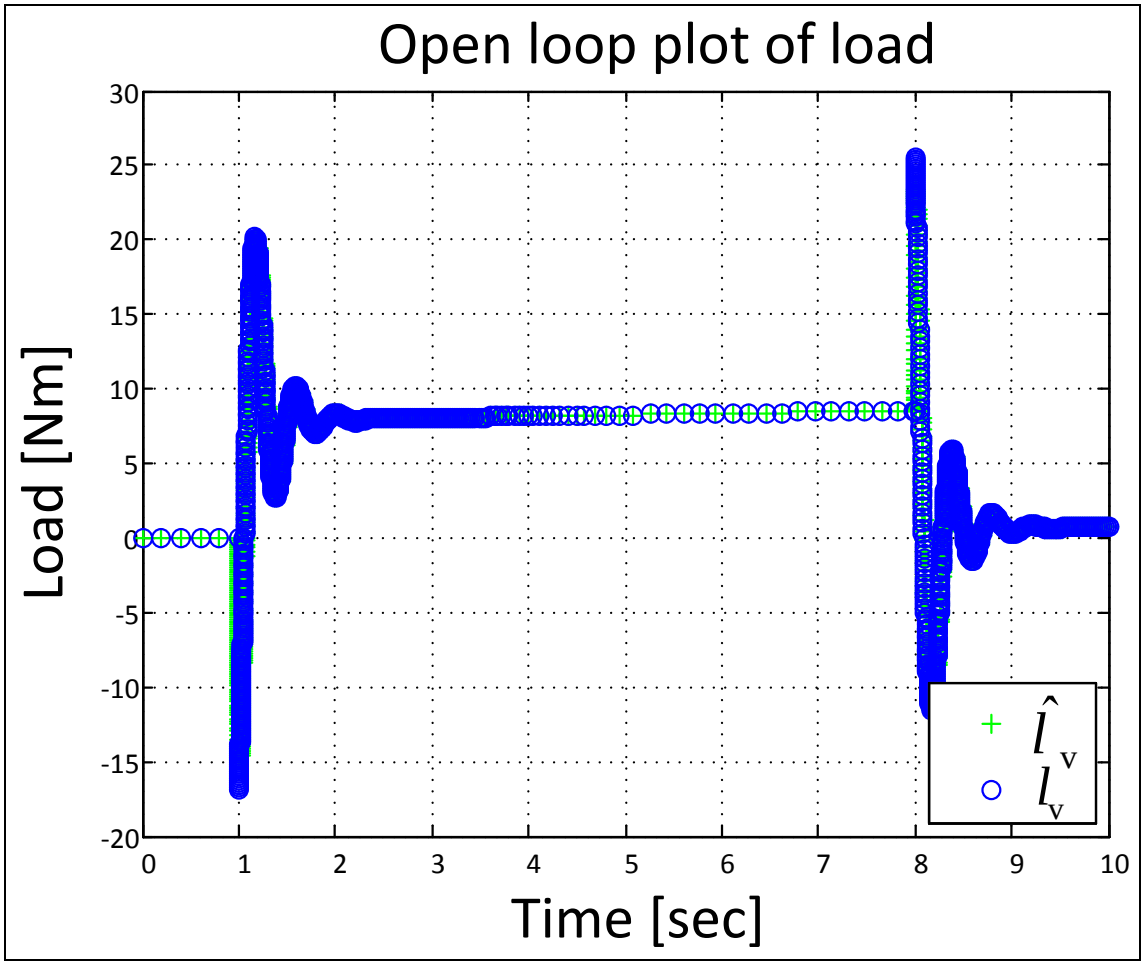


Figure 3-7: Plot of the actual load, l_v , and its estimate, $\hat{l}_v$, from the observer

Similarly, Figure 3.8 shows the ability of the observer to estimate the engine speed.

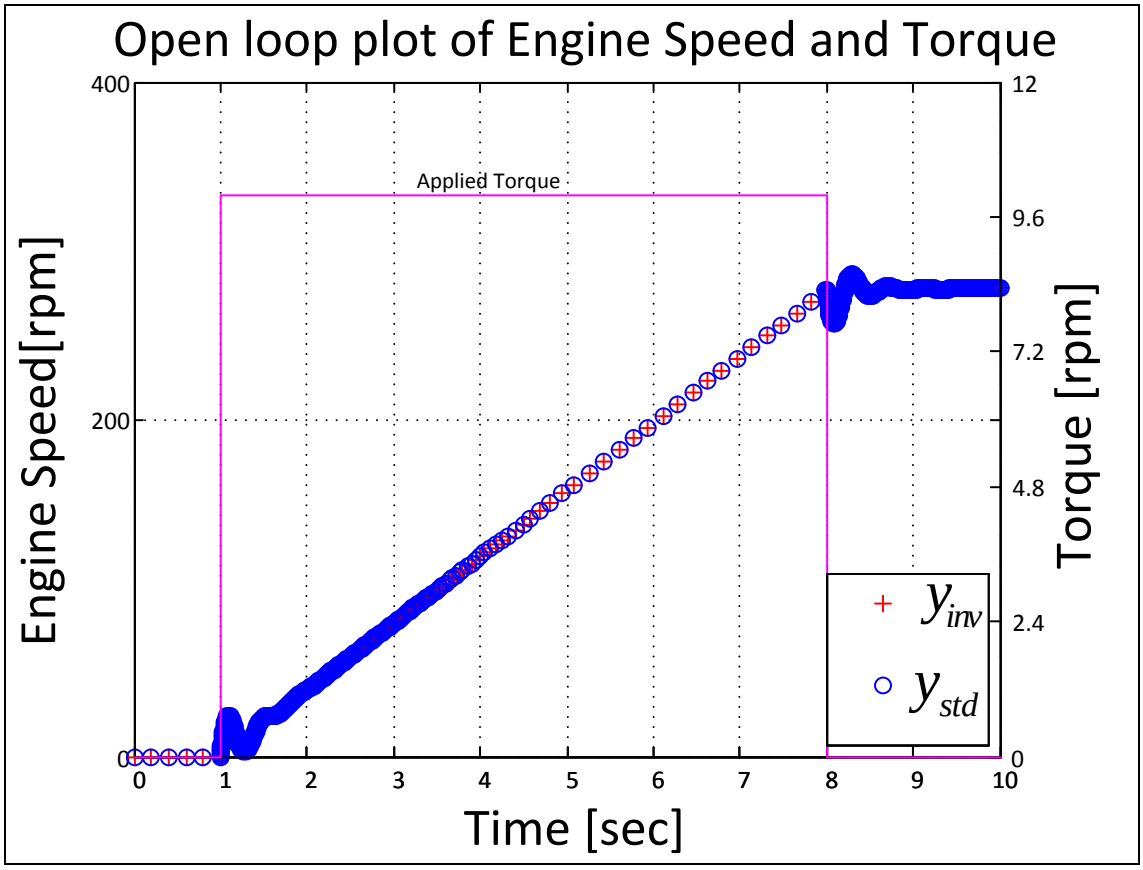


Figure 3-8 Engine speed, y_v, estimated engine speed, $\hat{y}_v$ and external load torque

Next the simulation results in the closed loop mode are presented. Here, $u(t)$ is produced by a simple proportional speed controller acting on the output of the inverse dynamic model, while the same input is applied to the standard model. The purpose of this simulation is to confirm that the two models are still equivalent under closed loop control and the observer continues to work correctly.

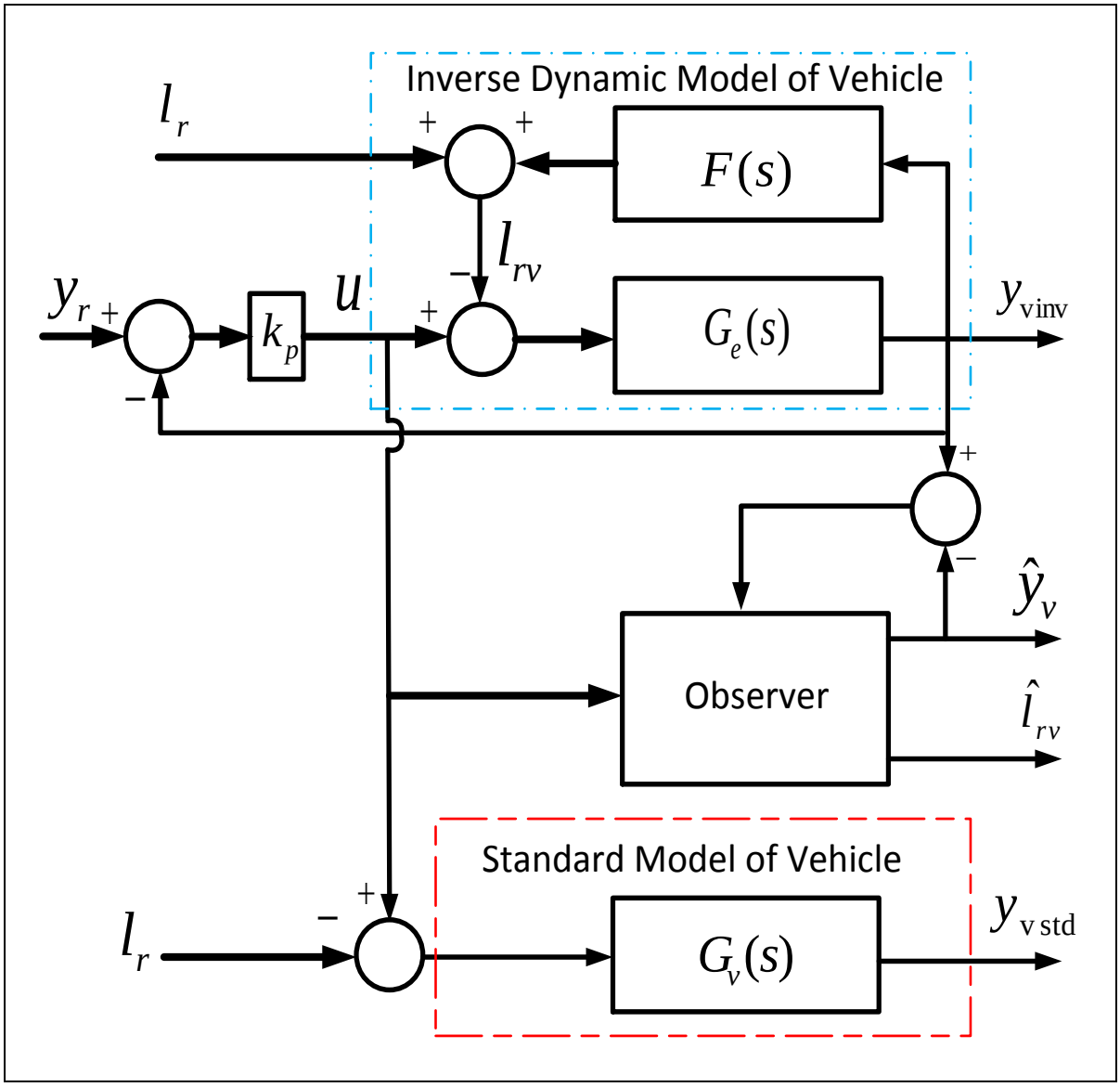


Figure 3-9: Standard and inverse dynamic vehicle models with proportional control loop

Figure 3-10 shows the closed loop response of the inverse dynamic model and the observer in the system of Figure 3-9. The proportional gain of the controller is set to $k_p = 0.1$ the load torque is cyclic so as to simulate the vehicle travelling over an undulating terrain.

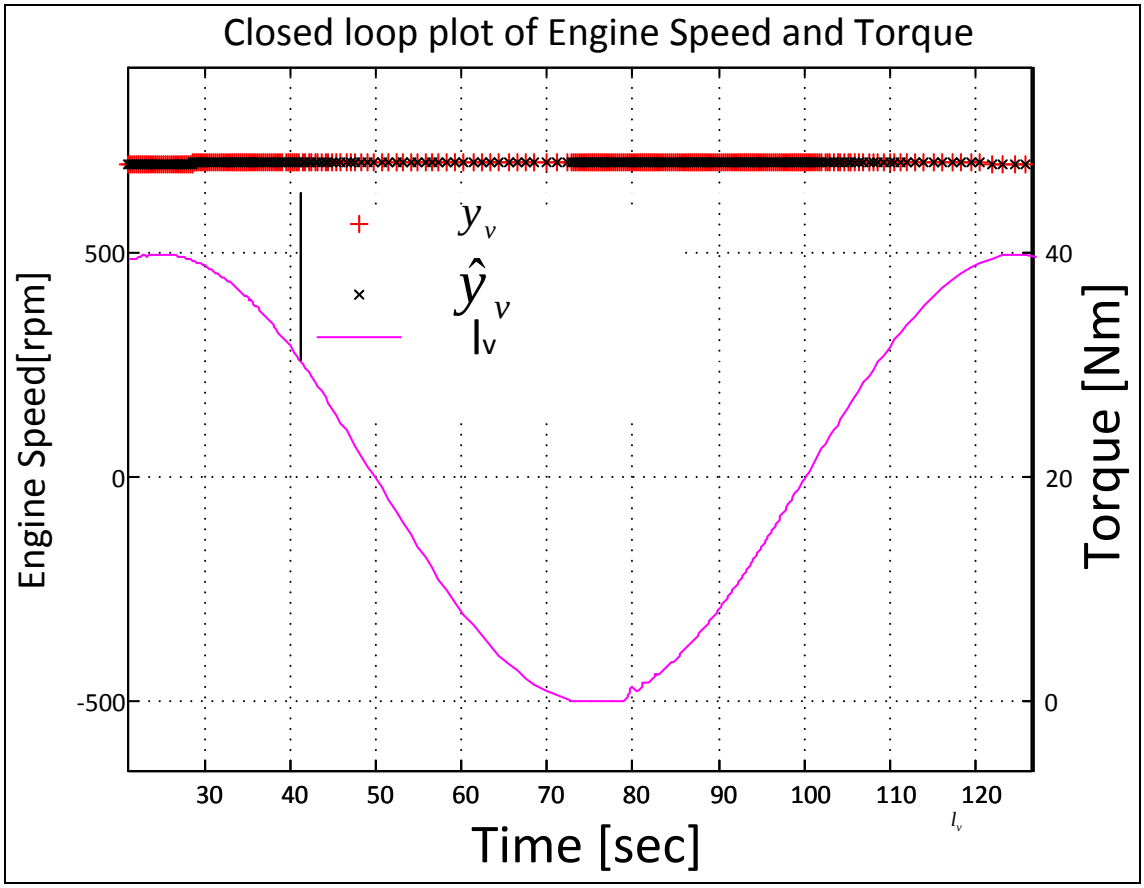


Figure 3-10: Plot of the estimated engine speed, $\hat{y}_v$, and the actual speed, y_v, with a cyclic external load torque, l_r

It can be seen that the estimated engine speed closely follows that of the inverse dynamic model. Remarkably, the simple proportional controller maintains a fairly constant speed.

Figure 3-10 shows the load applied at the flywheel and the estimated load by the observer model with the same piecewise constant external load applied at the engine flywheel as applied in the open loop mode above.

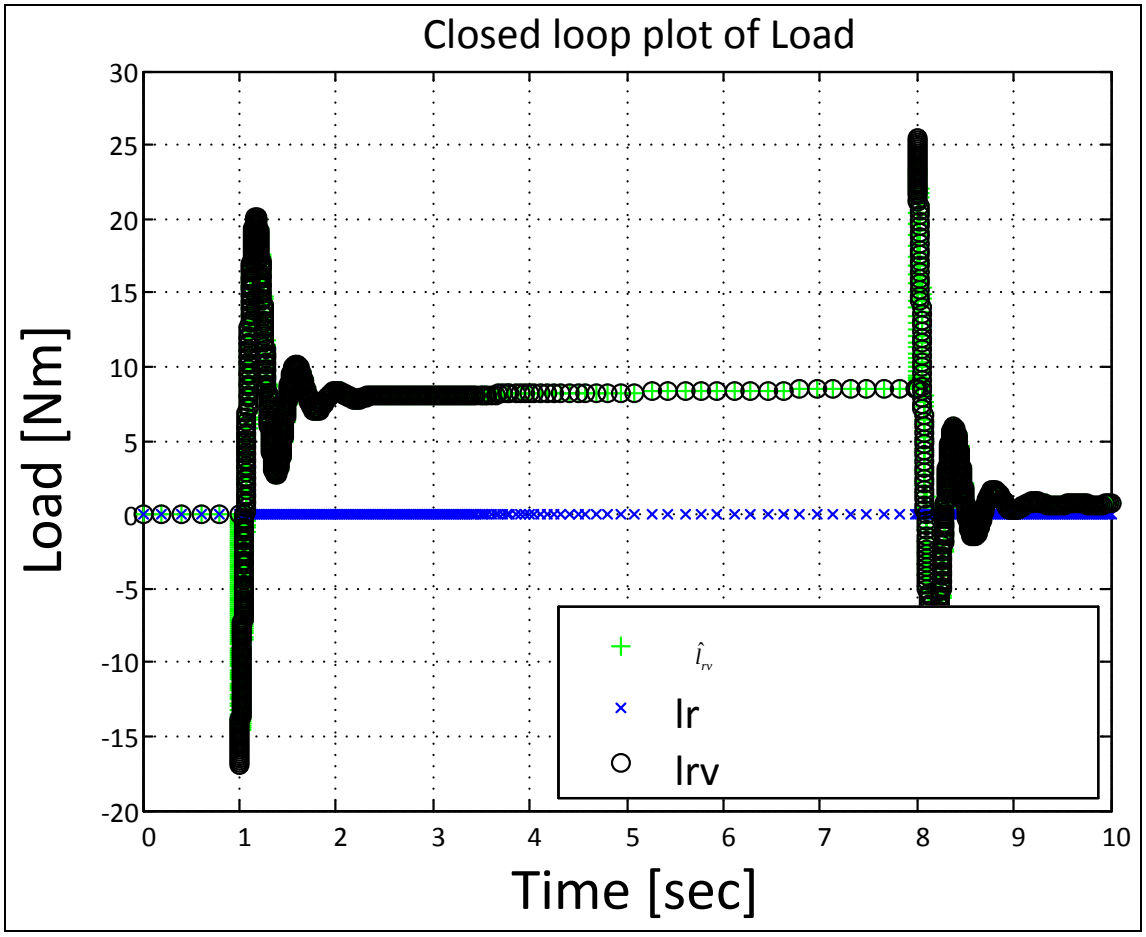


Figure 3-11: plot of the estimated load and inverse dynamic load

It can be seen that the estimation of this load is very good and is within an acceptable margin of error. During the transient condition, the observer estimation will be lagging the actual load. However, this transient estimation error is too small to be discernible in Figure 3-11 and is acceptable

3.6.3 Formulation of the unloaded Engine Model

An engine is a highly nonlinear system but can be approximated as a linear system with a very narrow band of linearity around the operating point. A simple physical model of the engine is used as a plant model. This is a torque proportional to the fuel flow rate, u, via a constant, K_e, called the engine gain, applied to a balanced body with moment of inertia, J_e, subject to a friction torque, Γ_r, that is proportional to the crankshaft angular velocity, ω, via a constant, B, that is a function of the oil and coolant temperature, Θ_o. The torque balance equation of this model is as follows:

$$J_e\dot{\omega} = K_e u - B(\Theta_o)\omega \tag{3.24}$$

For a given engine speed, the friction force will increase when the engine is cold and decrease as the engine warms up (cold to hot). The oil and coolant temperature of the engine, however, will have a DC effect on the frictional force in the sense that it will be sufficiently slowly varying to be considered constant on the time scale of operation of the control system. Then $B(\Theta_o)$ may be replaced by simply B in the equations of the model. The injection timing affects K_e but since this will be considered fixed during the control system operation, K_e is a constant plant parameter.. Figure 3.12 shows the engine block diagram based on the following re-arrangement of equation (3.24):

$$\dot{\omega} = \frac{1}{J_e}\left[K_e u - B\omega\right]$$

Taking Laplace transforms with zero initial conditions then yields

$$s\omega(s) = \frac{1}{J_e}\left[K_e u(s) - B\omega(s)\right] \Rightarrow \omega(s) = \frac{1}{J_e s}\left[K_e u(s) - B\omega(s)\right]$$

Figure 3-11 directly corresponds to this transfer function relationship.

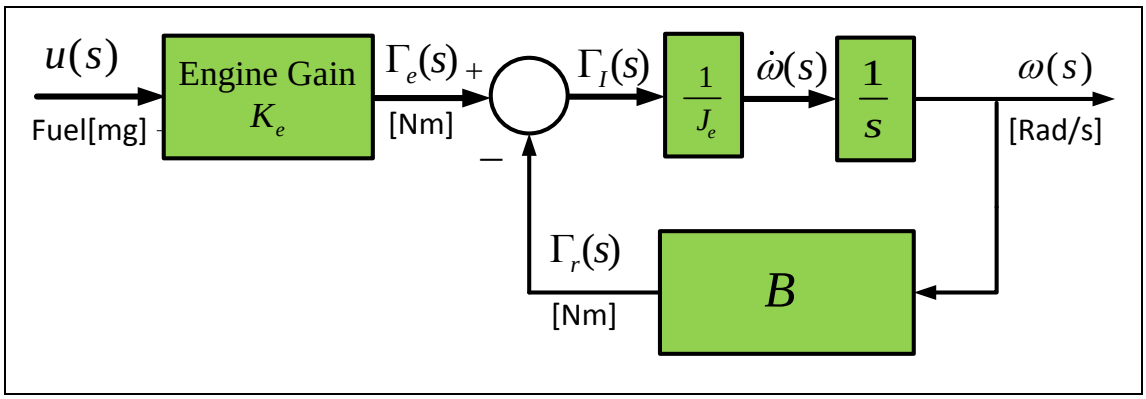


Figure 3-12: Diesel Engine Block diagram with Physical Parameters

The engine model may found in the form of a transfer function using this block diagram. Thus

$$G_e(s) = \frac{\omega(s)}{u(s)} = K_e \cdot \frac{\frac{1}{J_e s}}{1 + \frac{B}{J_e s}} = \frac{\frac{K_e}{J_e}}{s + \frac{B}{J_e}} = \underbrace{\frac{b}{s+a}}_{\text{Pole-zero form}} = \frac{\frac{K_e}{B}}{1 + s\frac{J_e}{B}} = \underbrace{\frac{K_{DC}}{1 + s\tau_e}}_{\text{Time constant form}} \tag{3.25}$$

The engine gain is the ratio of the maximum torque to maximum fuel injected at this torque. For a typical 10 liter 6 cylinder engine, this is

$$K_e = \frac{2800}{350} = 8[\text{Nm.Strk/mg}] \tag{3.26}$$

Table 3.1 shows typical engine data used in the simulations.

Table 3.1: Typical data for a 10 L 6 cylinder Diesel engine

Name	Values	Units
Capacity	10	L
Total moment of inertia, J_e	4.0	kgm^2
Friction torque at 500[rpm]	120	Nm
Friction coefficient, B		
Maximum torque	2800	Nm
Maximum fuel injection per stroke	350	Mg/stroke

Then the constant parameters of equation (3.25) are as follows.

$$b = \frac{K_e}{J_e} = \frac{8}{4.0} = 2 \text{ [strk/mg s}^2\text{]}$$

$$a = \frac{B}{J_e} = \frac{2.29}{4} = 0.57\left[\text{s}^{-1}\right]$$

$$K_{\text{DC}} = \frac{K_e}{B} = \frac{8}{2.29} = 3.49 \text{ [Nm.Strk/mg/s}^2\text{]}$$

and

$$\tau_e = \frac{J_e}{B} = \frac{4}{120} = 0.0\dot{3} \text{ [s]}$$

Figure 3-13 shows the block diagram of the observer with load torque estimation connected to the unloaded engine.

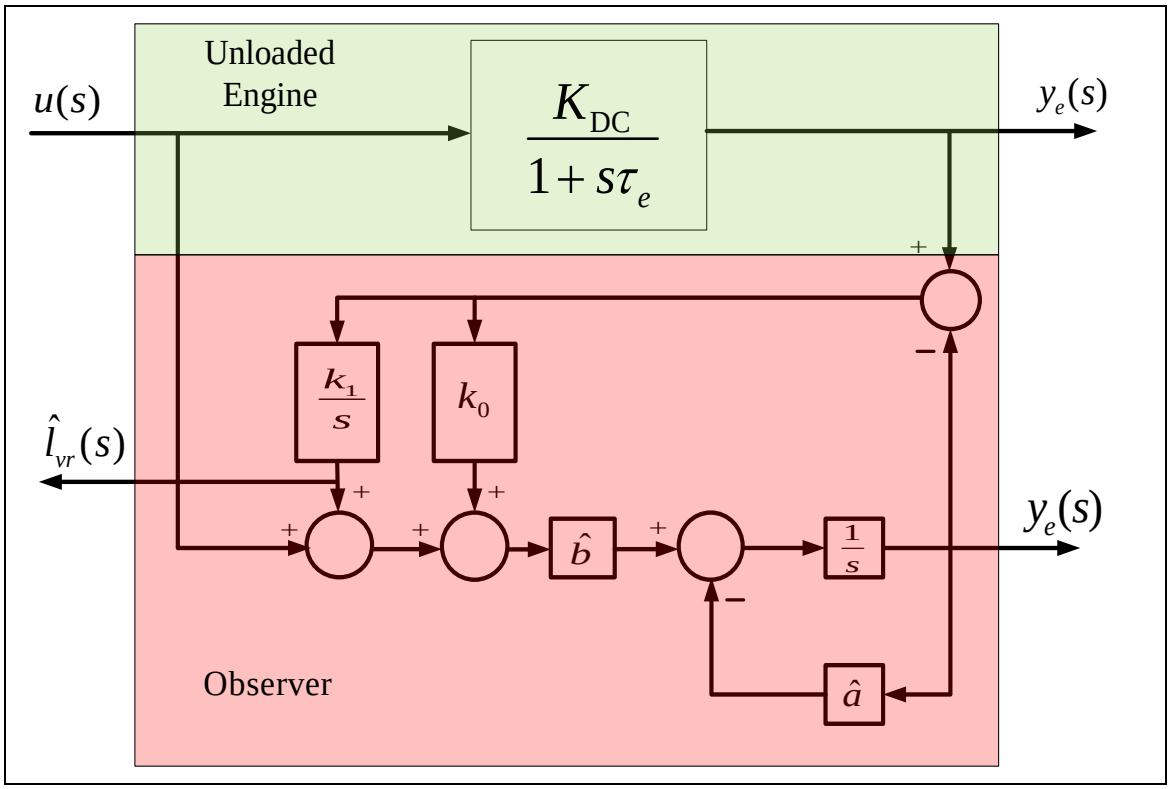


Figure 3-13: Block diagram of observer and unloaded engine

It should be noted that the controlled engine speed measurement is denoted $y_e(s)$ as in this case the engine is unloaded. This same variable, however, is denoted $y_v(s)$ when the engine is connected to the power train as in this case the whole vehicle is under control.

4 Observer Based Robust Control (OBRC): General Theory

4.1 Overview

Observer based robust control (OBRC) is a relatively new method [33]. This is based on an observer designed to estimate the equivalent disturbance input referred to the control input to a plant. The key to the method is that it represents plant modeling errors as well as external disturbances. This equivalent disturbance estimate is applied to the real plant input and the model of the plant in the observer to cancel its effect, thereby reducing the control problem to that of controlling the known real-time model of the plant employed in the observer. One of the disadvantages of conventional robust control methods such as those based on sliding mode control is that relatively high gain control loops are closed around the uncertain plant, thereby risking instability. The original motivation for considering OBRC is the fact that the high gain loops in the system were apparently restricted to the observer. Thus, applying it to a known plant model avoids any risk of instability. As will be seen, a step in the formation of an OBRC controller will convert the problem of controlling an uncertain plant to the simple one of controlling the known model in the observer. This step, however, also effectively transfers the high gain loops of the observer to the unknown plant. Despite this, OBRC is included in view of its unconventional structure. Due to this, different results may be expected than those from other control techniques, such as robust pole placement, which directly close high gain loops around the uncertain plant. The method is demonstrated via its application to Diesel Drivelines for commercial road vehicles.

One of the operational problems with conventional PI engine speed controllers is the need for time consuming initial controller tuning, requiring different sets of gains for each gear selection, including idle (i.e., neutral) and later retuning to compensate for changes in the driveline characteristics with component ageing. A major advantage of OBRC in this application is the elimination of the tuning procedure. Also, of particular interest is the fact that the order of the system is increased by two when a gearbox is considered in the transfer function of the system.

Observer based robust control (OBRC) is a new control technique, which is applicable to linear or nonlinear uncertain plants subject to unknown disturbances. It achieves robustness according to the following definition: The robustness of a control system is defined as its ability to produce a specified closed loop dynamic response to reference inputs, within acceptable error tolerances for the application in hand, despite:

1. Uncertainties in the assumed plant model used for the control system design.

2. Unknown external disturbances.

A specified closed loop dynamic response means that the output response to a given reference input is determined by a specified differential equation. The error tolerances are included to allow acceptably small departures from the ideal closed-loop response. An 'uncertain plant' refers to a plant whose mathematical model is not known accurately: In addition the lack of accurate knowledge of the plant may include uncertainty of the plant order. In a mechanical system, this is due to the elastic components of a controlled mechanism having an unlimited number of vibration modes, each contributing two to the order, while the controller design has to be based on a model of finite order.

In the following sections, the OBRC control technique is developed using a general multivariable plant model,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{d}),\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{4.1}$$

where $\mathbf{x} \in \Re^n$ is the state vector, $\mathbf{u} \in \Re^m$ is the control vector, $\mathbf{d} \in \Re^m$ is an external disturbance vector referred to the control input, $\mathbf{y} \in \Re^m$ is the measurement vector and the vector functions, $\mathbf{f}(\bullet)$ and $\mathbf{h}(\bullet)$, are continuous. It is later applied to the single input, single output (SISO) Diesel driveline application.

4.2 The 'plant model mismatch equivalent input' premise

The applicability of the OBRC method depends on the existence of a plant model mismatch equivalent input, $\mathbf{u}_e$, the meaning of which is defined in Figure 4.1

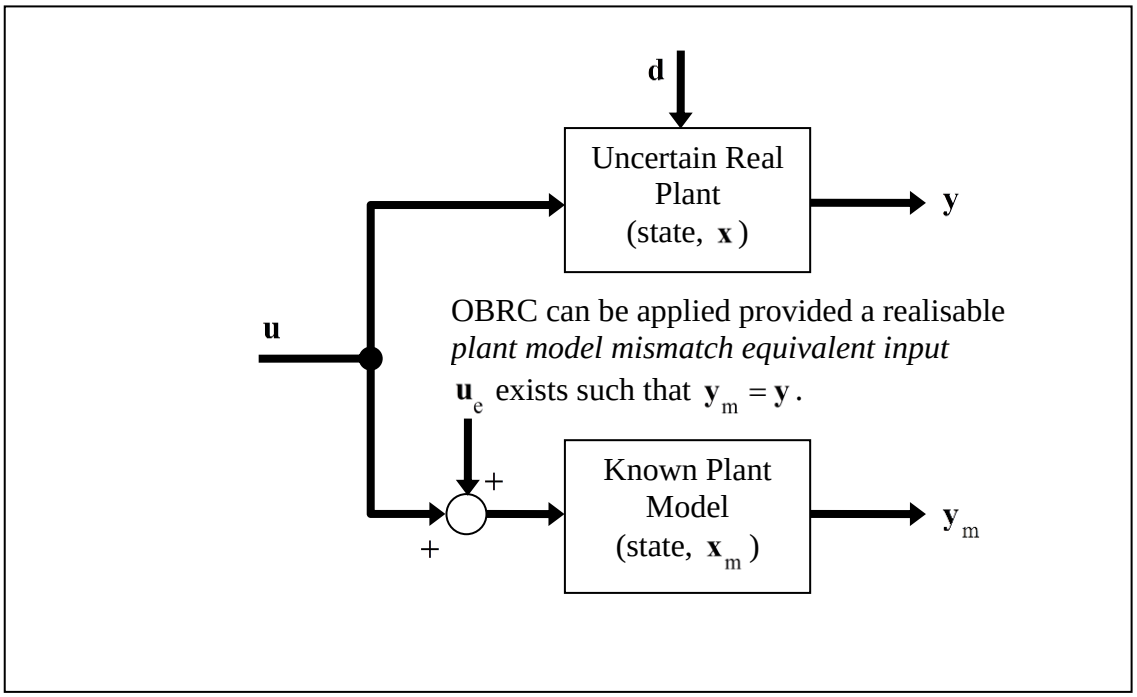


Figure 4-1: The plant model mismatch equivalent input

It is important to realise that OBRC is only realisable without control actuator saturation, i.e.,

$$\mathbf{u}_{min} \le \mathbf{u} \le \mathbf{u}_{max} \tag{4.2}$$

4.3 Formation of controller using the plant model mismatch equivalent input

If $\mathbf{u}_e$ were to be known, then the control input could be formed as $\mathbf{u} = \mathbf{u}' - \mathbf{u}_e$, as shown in Figure 4.2 (a). Then $\mathbf{u}'$ would become the direct control input for the plant model, as shown in Figure 4.2(b).

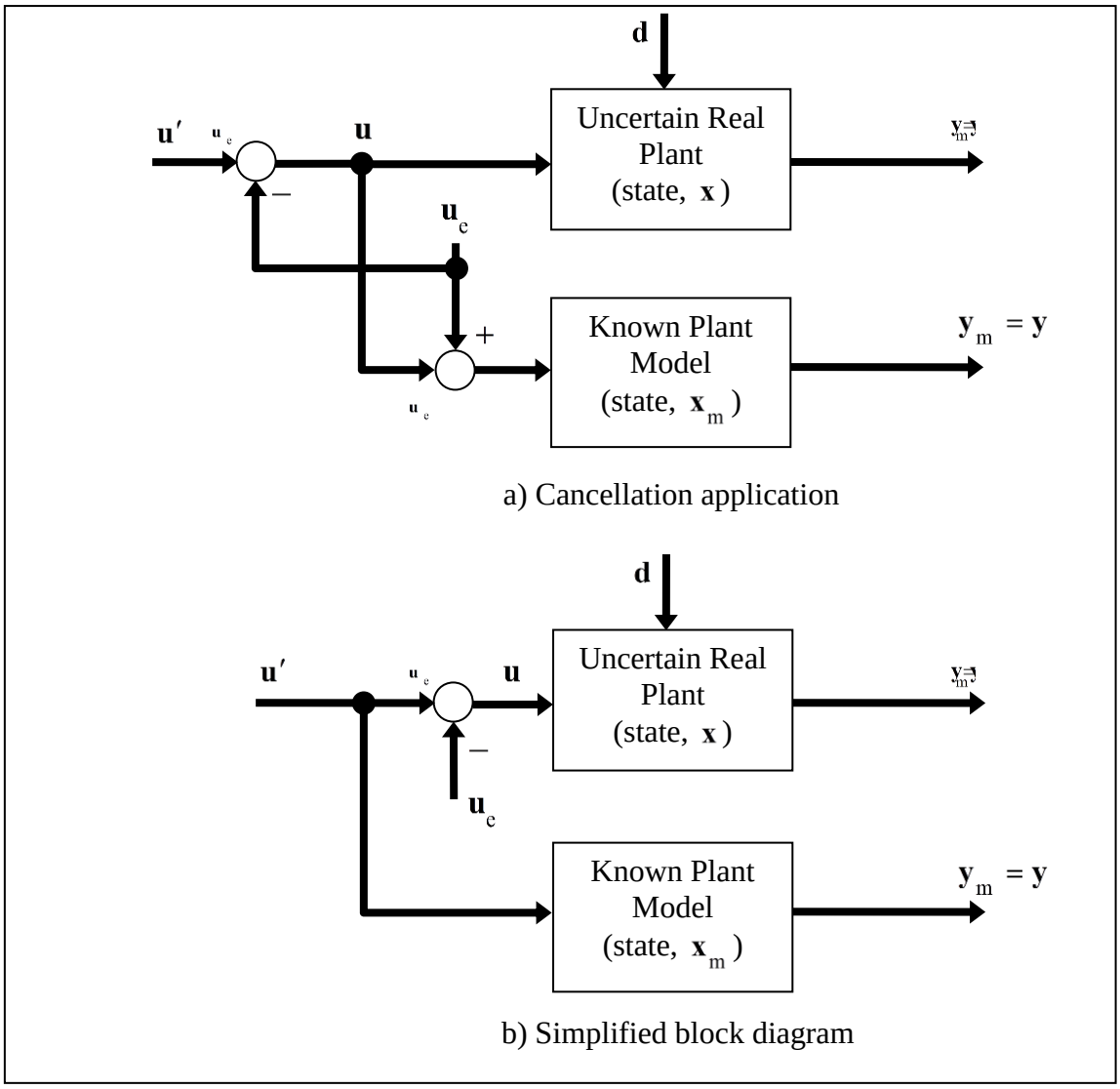


Figure 4-2: Plant model mismatch input cancellation

A state feedback controller could be designed for the known plant model which would control the real plant as $\mathbf{y} = \mathbf{y}_m$ as shown in Figure 4.3.

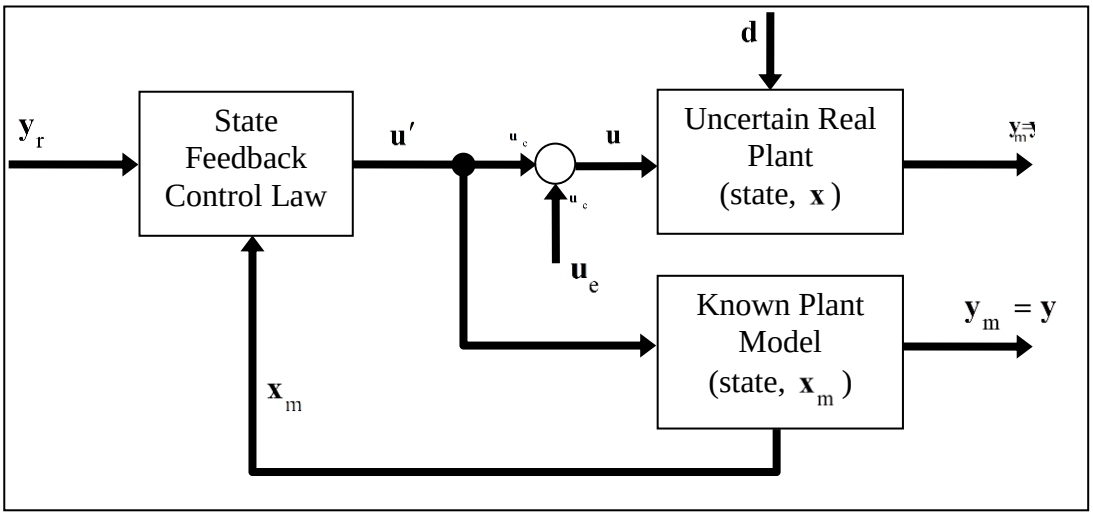


Figure 4-3: Controlling the uncertain real plant via control of the plant model

What is needed to render this method practicable is an estimate of $\mathbf{u}_e$ and this is provided by an observer as shown in the following section.

4.4 Introduction of an Observer for the Estimation of External Disturbance

In Figure 4.3, a loop is closed around the known real time plant model, the purpose of which is to drive the error between $\mathbf{y}_m$ and $\mathbf{y}$ to negligible proportions. Note that for the formulation of an observer required to estimate $\mathbf{u}_e$, it is necessary to revert to Figure 4-2 (a), since the plant control input, $\mathbf{u}$, has to be applied to the known plant model. A variant of the observer, however, will be presented subsequently that permits the simplification depicted in Figure 4-2 (b) to be carried out again.

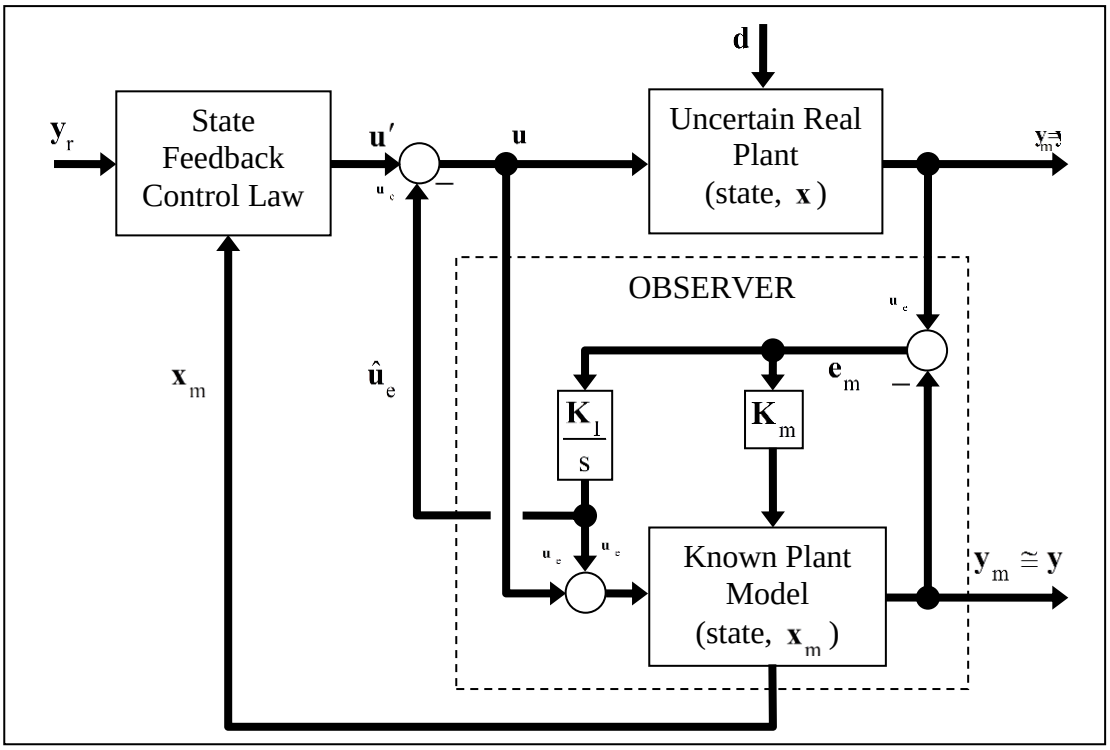


Figure 4-4: Introduction of an observer

It is evident by comparing Figure 4.4 with Figure 4.2 (a) that if the observer gains are made sufficiently large for $\mathbf{e}_m \cong \mathbf{0}$ then $\mathbf{y}_m \cong \mathbf{y}$. The model state, $\mathbf{X}_m$, is readily available and may be used for control purposes. It is well known that in a conventional observer that the plant model has to be as accurate as possible. This is so that the model state used to calculate the controller's input is closely following the real plant state, thereby enabling a close approach to ideal state feedback to be obtained. This implies that the real plant order has to be known. Quite remarkably however, if the correction loop is 'tight' enough to keep the error, $\mathbf{e}_m$ to negligible proportions, then it would appear that this normally accepted restriction may be removed. In this case neither the plant order nor the plant parameters would have to be known, leaving the possibility of obtaining extreme robustness, but it is possible for the system to contain internal unstable modes even with $\mathbf{e}_m \cong \mathbf{0}$ and each case in which the model order differs from the plant order must be carefully examined and simulated to identify such modes if they exist.

It will also be noticed that in Figure 4.4, the model correction loop includes a path through the gain matrix, $\mathbf{K}_m$, that bypasses the integrators in the outer part of the loop that produce $\hat{\mathbf{u}}_e$. To avoid any possibility of losing information about the true $\hat{\mathbf{u}}_e$ in this way, a modified observer may be formed in which the model correction is applied only at the input of the model, in contrast to conventional state variable block diagrams of observers in which the correction loop is implemented by applying the error to the input of every first order subsystem of the model via a pre-calculated gain. It may be easily shown, however, that any conventional observer may be represented in this form. Figure 4.5 shows the block diagram.

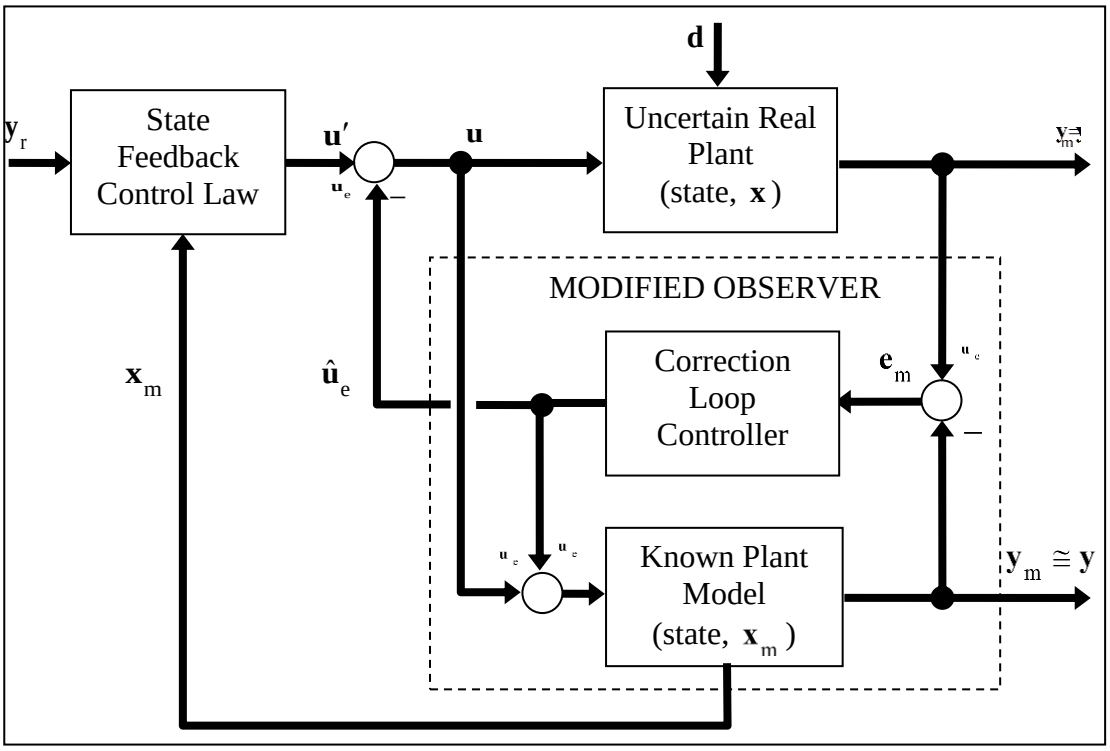


Figure 4-5: Introduction of modified observer

It is important to note that the settling time of the correction loop has to be chosen at least an order of magnitude smaller than that of the main control loop controlling $\mathbf{y}_m$ and $\mathbf{y}$ in order to drive $\mathbf{e}_m$ to small proportions. This necessarily gives the correction loop controller relatively high gains.

The modified observer also permits the simplification of the block diagram carried out in Figure 4.2 (b), resulting in Figure 4-6.

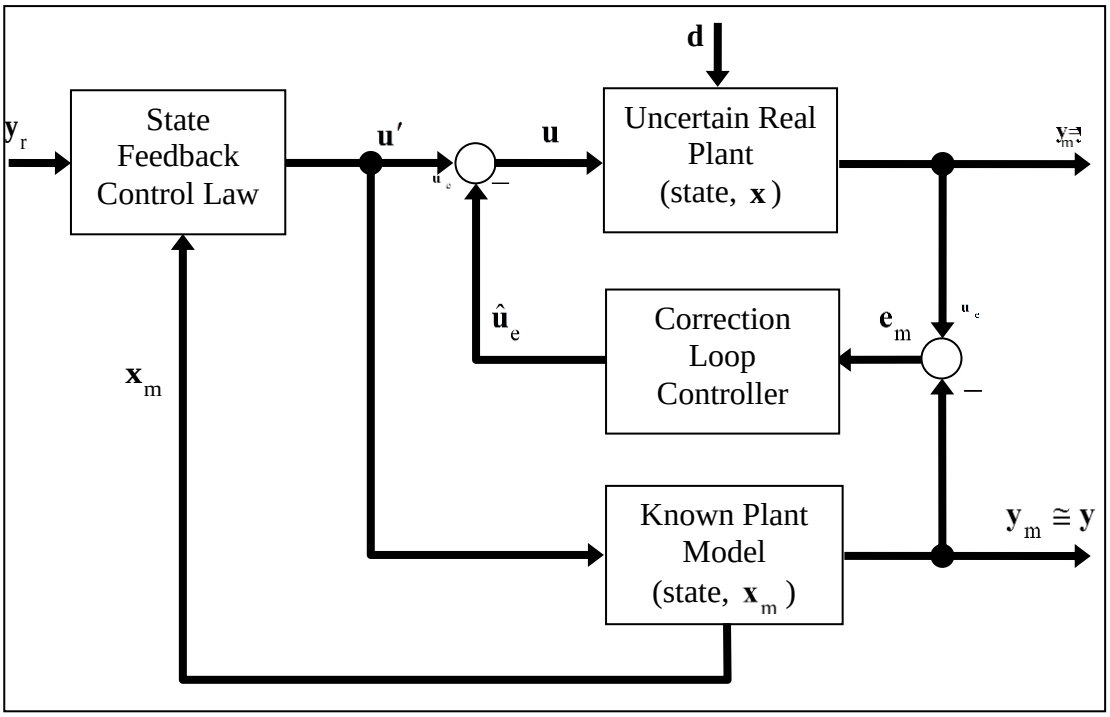


Figure 4-6: Simplified form of OBRC based on modified observer

4.5 Application to Diesel Engine Control

In this section, the variants of the OBRC developed in the previous sections are applied to the same plant as considered in subsection 3.6.4. Figure 4-7 shows the basic OBRC using the observer of section 3.5, which is a particular case of the system shown in Figure 4-4. Figure 4-8 shows the OBRC using the modified form of this observer, which is a particular case of the system shown in Figure 4-5. It should be noted that in this case the correction loop controller has *precisely* the same effect as the integrator and proportional blocks of the conventional observer in Figure 4-7. Figure 4-9 shows the block diagram resulting from the simplification equivalent to that leading to Figure 4-6. Finally, since only the model output, $\hat{y}_e$, is needed for the model control, this may be replaced by the available measurement, y_e, as shown in Figure 4-10.

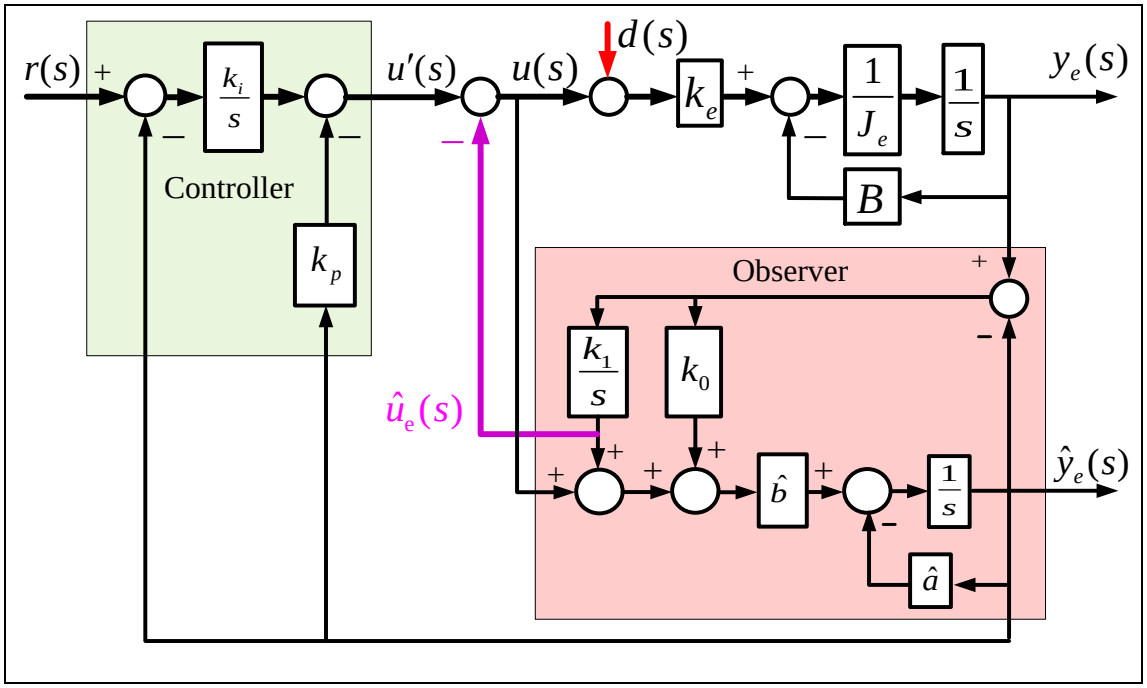


Figure 4-7: Basic OBRC of Diesel engine

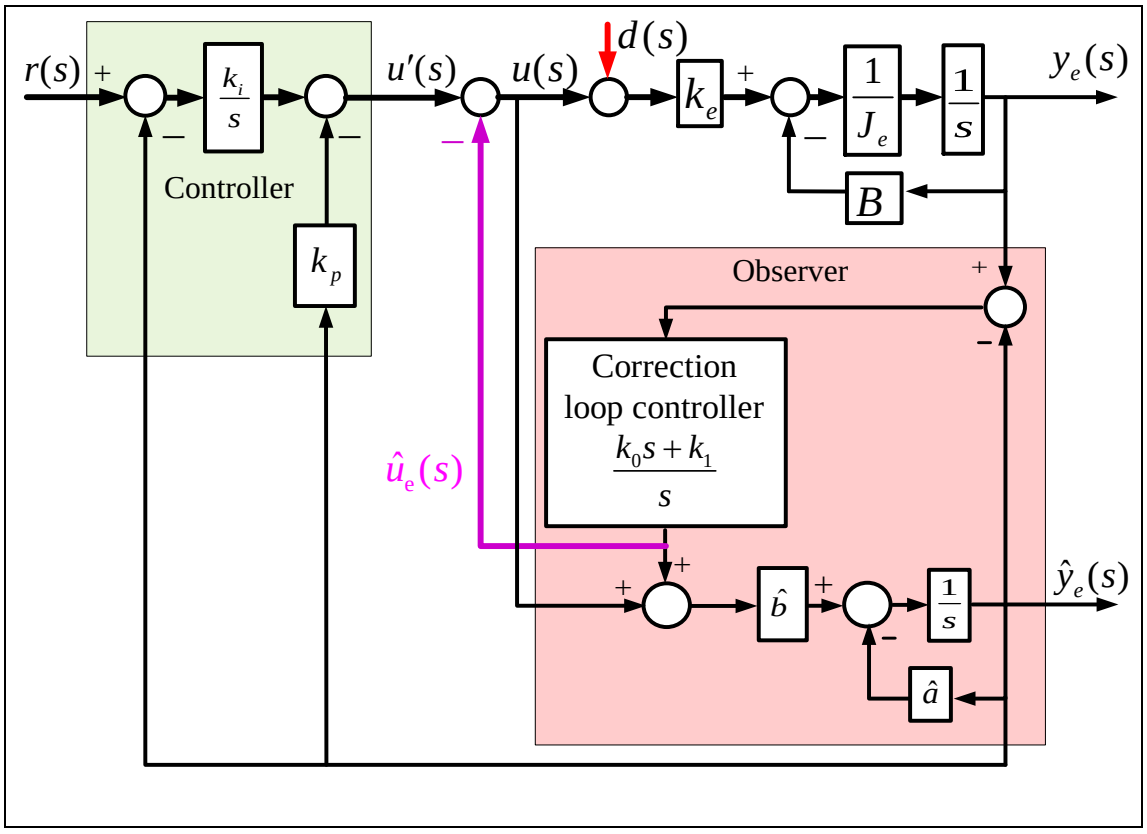


Figure 4-8: OBRC of Diesel engine using modified observer

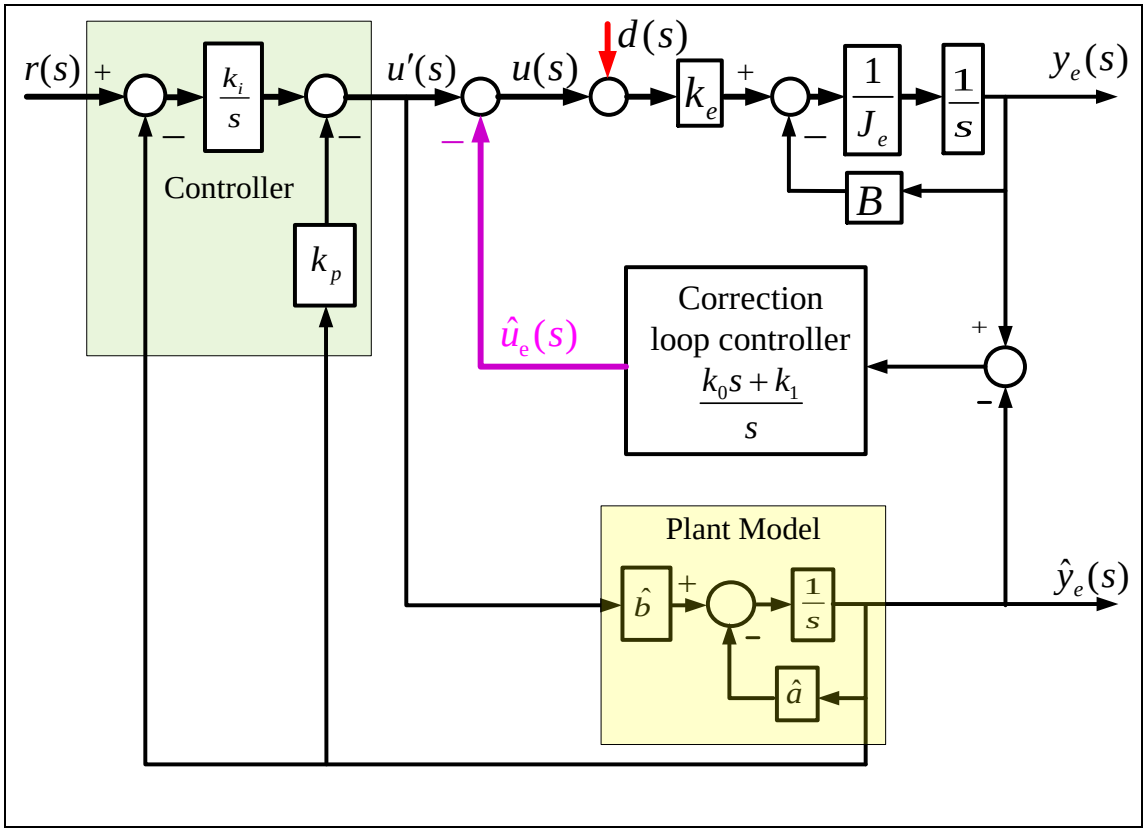


Figure 4-9: Simplified OBRC of Diesel Engine

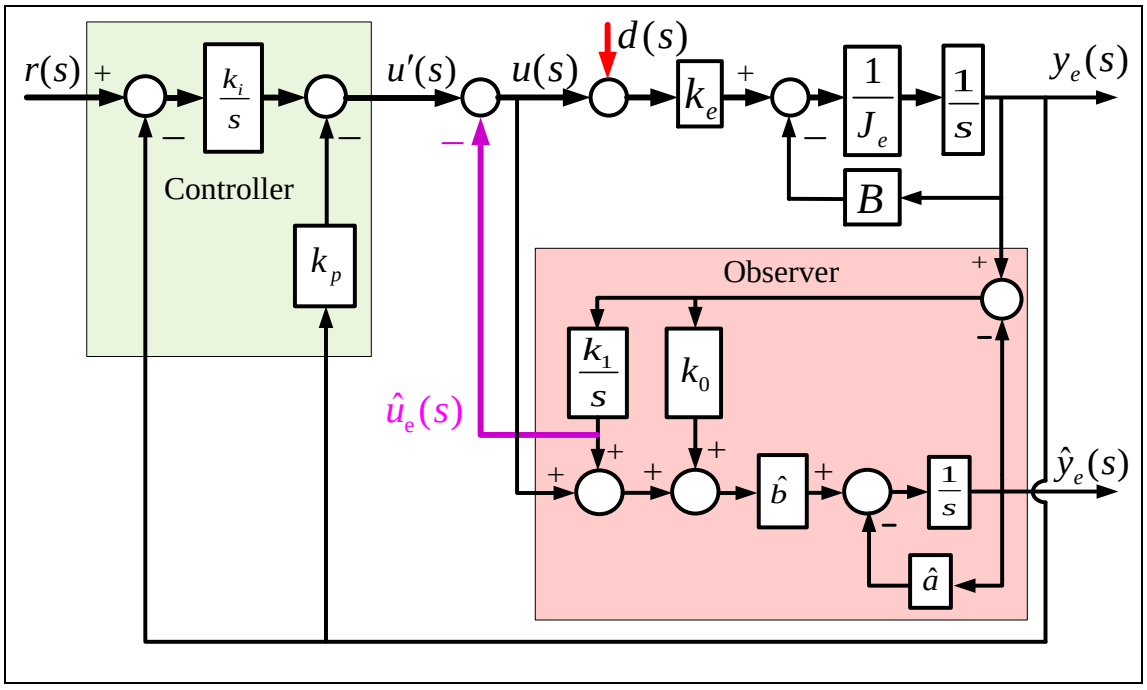


Figure 4-10: Basic OBRC of Diesel engine with direct plant control

5 Comparison of Control Techniques for General Applications

5.1 Overview

This chapter concentrates on the comparison of all the control techniques to be considered using various plant examples. Further comparisons will be made specifically for the Diesel driveline application in Chapter 6. The main aim is robustness comparison in view of its advantages in the engine application. The OBRC controller [10, 20] is important from this viewpoint as it has been produced specifically for robustness. The RHF controller with robust pole assignment is also considered a strong contender. H-infinity [15, 30] is included as it has also been produced for robustness. This, together with the IMC controllers has also been selected, because there is off-the-shelf design software available for these control techniques. Finally the RST controller is included as it is surfacing as a mainstream control technique and is identical in structure to the RHF controller evolved independently by the author's supervisor but with an entirely different design procedure based on pole placement. A range of plants have been selected for this comparison as they are common examples used in academia as well as applications in industry. The comparison includes the effort needed in the design process and the tools that one has to employ.

5.2 Introduction to Other Control Techniques and their Design Procedures

In recent years there have been developments of controller design based on the specification of the desired closed loop transfer function, which is a form of pole placement. This design technique applies to several different control techniques that yields a linear closed loop system. The pole assignment is an obvious choice, but has not been used in the past due to the number of adjustable controller parameters being limited by traditional analogue electronic circuit implementation. This restriction no longer applies. In recent years there has been some further development in this area which provides a stepwise robust control procedure in designing a control algorithm

[37]. The type of controller to which this applies is known as the RST-Controller [13], the acronym simply referring to the symbols used in its transfer function blocks. There are a number of control methodologies that have been developed, with papers being written on the industrial process control [36]. In general, the controllers are designed in the z-domain in view of the digital implementation. Polynomial control, however, has been developed in the Laplace domain but can easily be adapted to the z-domain.

The three plant transfer functions used for the comparisons are as follows:

$$P_1(s) = \frac{-1(-s+1)}{(3s+1)^3} \tag{5.1}$$

$$P_2(s) = \frac{400}{s^2 + 2s + 400} \tag{5.2}$$

$$P(z^{-1}) = \frac{2z^{-1}(1+2z^{-1})}{(1-z^{-1})(1-0.3z^{-1})} \tag{5.3}$$

In this chapter, a brief introduction to the RST-controller is provided and this is assessed using plant (5.3). The pole placement based polynomial control design methodology used for the RHF controller will also be applied. Here, two plants will be considered, one with transfer function (5.1) in view of its RHP zero as shown by equation references [19] and the other with transfer function (5.2) that has no RHP zero. The closed loop performances will be compared with that of the polynomial controller using Dodds' time domain design procedure based on his settling time formula (2.23).

In the following sections, the design of each of the controllers is presented including the OBRC.

5.3 OBRC Controller Design

5.3.1 Overview

The design process is divided into two parts, one is the design of the observer and the other is the design of the model state controller. These parts are presented in the following two sections.

5.3.2 OBRC Model State Control Loop

Figure 5.1 shows the OBRC block diagram with the plant (5.2).

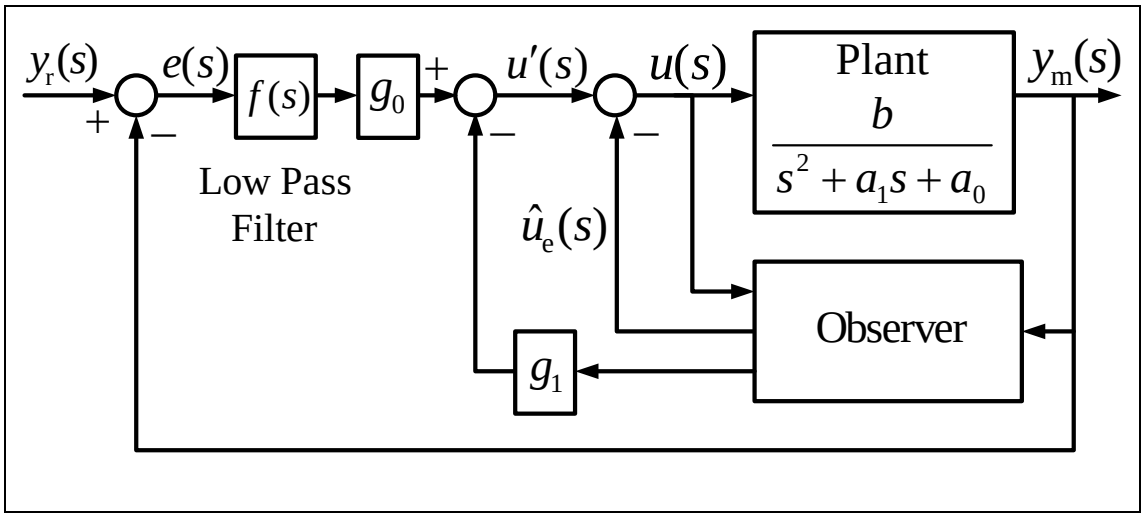


Figure 5-1: OBRC closed loop system showing details of model state control loop

Figure 5.2 shows the model state control loop of the OBRC block diagram isolated from the rest of the control system for the purpose of calculating the state feedback gains, g_1 and g_0.

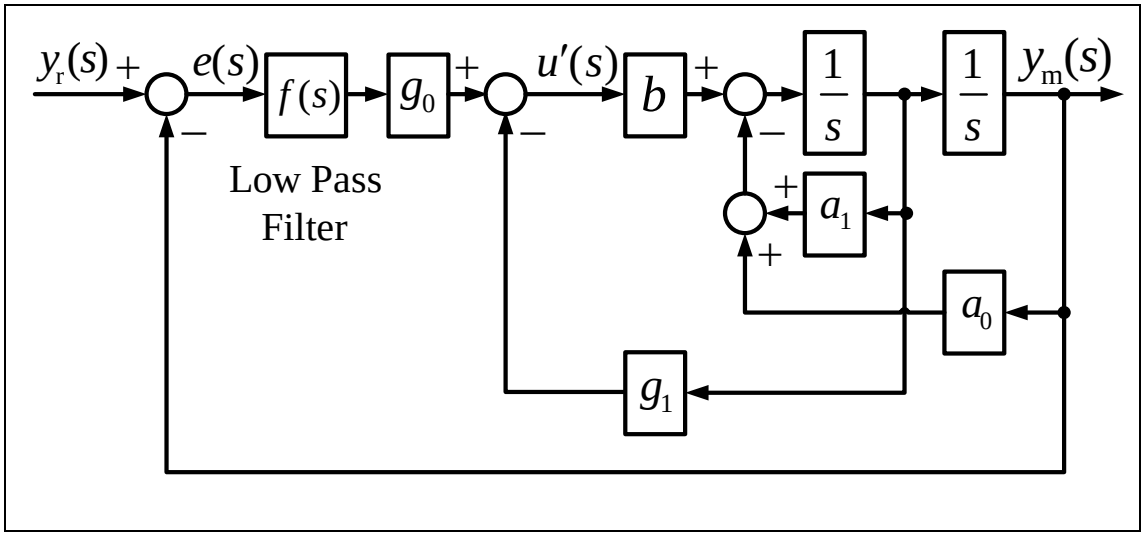


Figure 5-2: Model state control loop of OBRC for gain calculation

The low pass filter with transfer function,

$$f(s) = \frac{K_f}{(T_f s + 1)} \tag{5.4}$$

is included to avoid high frequency components of measurement noise in the control signal and also to reduce the controller effort at high frequency. However introducing the filter into the block will affect the closed loop performance. On the other hand, it will increase the robustness against the model uncertainty. The filter time constant, T_f, however, is an additional adjustable parameter that could be used for pole assignment to achieve a specified third order transient response. On the other hand, if T_f can be less than the settling time, T_s, by at least an order of magnitude, then the gains could be calculated for the second order system obtained by replacing the low pass filter by its DC gain, K_f. This was set to 0.67 to bring the low frequency gain to a unity. Rather than undertake the third order pole placement calculation, of the filter time constant was set to $T_f = 0.03[\text{s}]$ which yields a cut-off frequency of the same order as the closed loop design specification to ensure the controller effort is limited at high frequencies and thereby prevent actuator chattering.

The closed loop transfer function of the system in Figure 5-1 is

$$\frac{y(s)}{y_r(s)} = \frac{f(s)g_0 b}{s^2 + (bg_1 + a_1)s + f(s)g_0 b} \tag{5.5}$$

The desired characteristic equation was determined by replacing $f(s)$ by K_f in transfer function (5.4) using equation 2.23 as the desired characteristic equation with $n = 2$ and a settling time of 0.04 [s] was chosen. Thus the desired closed loop characteristic polynomial was determined as

$$D_{eq}(s) = s^2 + 225s + 1.266 \times 10^4 \tag{5.6}$$

Now, g_1 and g_0, are determined by equating the corresponding coefficients of the denominator polynomial of equation (5.5) and (5.6). Note for this calculation the low pass filter is $f(s) = 1$. Therefore

$$g_1 = \frac{225 - a_1}{b} = \frac{(225 - 2)}{400} = 0.5575 \tag{5.7}$$

$$g_0 = \frac{1.266 \times 10^4}{400} = 31.65 \tag{5.8}$$

The effect of filter is shown in the figure 5.4(d).

5.3.3 Determining the OBRC Observer Gains

Figure 5-3 shows the observer block diagram.

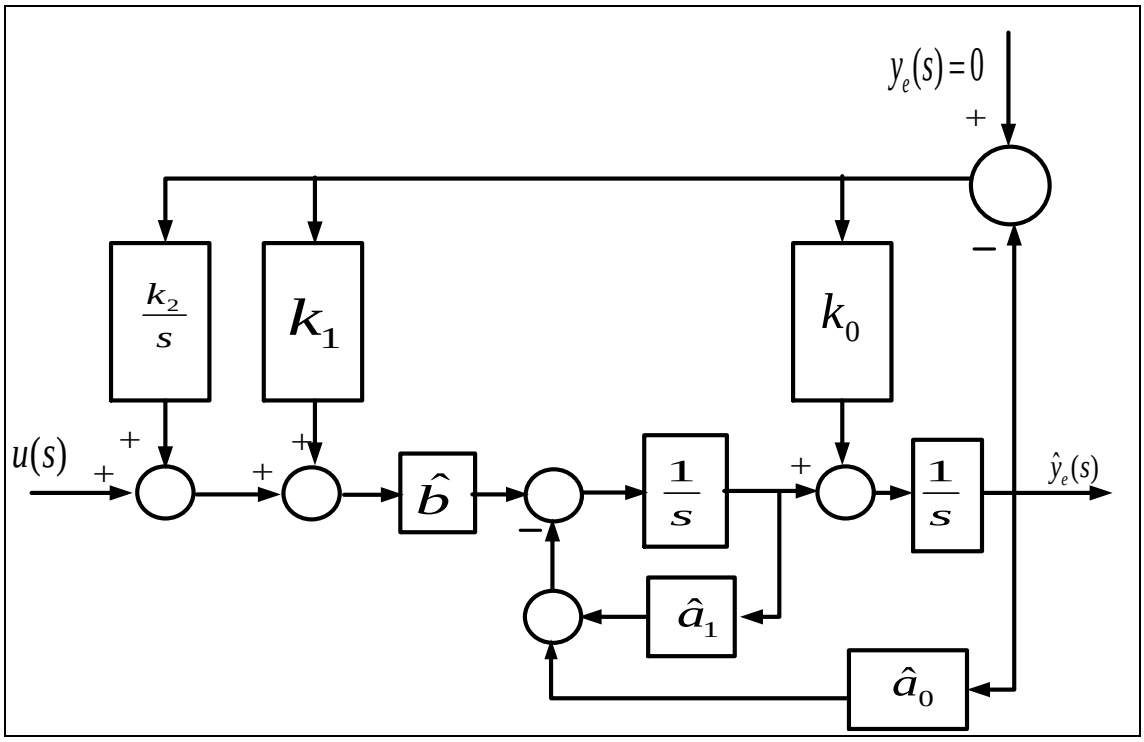


Figure 5-3: OBRC observer block diagram

The characteristic equation of the observer is given by equating the determinant of Mason's formula to zero. Thus

$$C_{eq} = s^3 + (\hat{a}_1 + k_0)s^2 + (k_0\,\hat{a}_1 + \hat{a}_0 + k_1\,\hat{b})s + \hat{b}k_2 \tag{5.9}$$

The desired characteristic equation for the observer is determined by using the settling time formula 2.23 and since the system is 3^{rd} order system i.e. $n = 3$. Hence

$$D_{eq} = s^3 + 6750s^2 + 1.519\times10^7 s + 1.139\times10^{10} \tag{5.10}$$

To determine the coefficient of the observer controller k_2, k_1 and k_0, equate the coefficient of the equation 5.9 and 5.10. Hence

$$k_2 = \frac{1.139\times10^{10}}{\hat{b}} = \frac{1.139\times10^{10}}{400} = 28475000 \tag{5.11}$$

$$k_1 = \frac{1.519\times10^{7} - (\mathrm{k}_0\,\hat{a}_1 + \hat{a}_0)}{\hat{\mathrm{b}}} = \frac{1.519\times10^{7} - (1.519\times10^{7}\times2 + 400)}{400}$$

$$k_1 = \text{-}37976 \tag{5.11a}$$

$$k_0 = 1.519\times10^{7} - \hat{a}_1 = (1.519\times10^{7} - 2) = \ 15189998 \tag{5.12}$$

5.3.4 Cascade Controller equivalent to OBRC

To allow us to compare the OBRC controller in the frequency domain, we will determine the OBRC controller by creating an equivalent closed loop block diagram with unity feedback. Hence

$$G_{cl}(s) = \frac{(K_{obrc}P(s))}{(1+K_{obrc}P(s))} \tag{5.13}$$

where $K_{obrc}(s)$ Is the equivalent transfer function of the OBRC controller and $P(s)$ is the plant with transfer function given by equation 5.2. Re-arranging (5.9) and making the "K_{obrc}" the subject yields

$$K_{obrc}(s) = \frac{(G_{cl}(s))}{(P(s)+G_{cl}(s)P(s))} \tag{5.14}$$

Therefore

$$L_o(s) = K_{obrc}P(s)$$

$$S_o = \frac{1}{(1+L)}$$

$$T_o = \frac{1}{(1+S_o)}$$

Hence, the following equations are derived from the closed loop transfer function using the above relationship.

$$K_{obrc}(s) = \frac{282656.25(s^3 + 9002s^2 + 180400s + 3.6\times10^{7})}{(s^4 + 1.8\times10^{5}s^3 + 8.106\times10^{9}s^2 + 2.7\times10^{11}s + 1.2\times10^{9})} \tag{5.15}$$

$$S_o = \frac{(s+150)(s+9\times10^{4})^2(s+0.004444)(s^2+2s+400)}{(s+9\times10^{4})^2(s^2+149.9s+5626)(s^2+2s+400)} \tag{5.16}$$

$$T_o = \frac{-1.0904\times10^{-11}(s-5.452\times10^{9})(s+5.452\times10^{9})(s+7.2\times10^{4})(s^2+2s+400)}{((s+7.2\times10^{4})^2(s+108.4)(s+41.5)(s^2+2s+400))} \quad (5.17)$$

The closed loop transfer functions of the system for the OBRC and H-infinity is given by equations (5.18) and (5.19) respectively

$$G_{clobrc} = \frac{202499999.9997}{(s+9\times10^{4})(s^2+61.98s+2250)} \quad (5.18)$$

$$G_{H\inf} = \frac{119092918.5615(s+2.095\times10^{4})(s+6200)}{(s+2.095\times10^{4})^2(s+6199)(s^2+146.4s+5737)} \quad (5.19)$$

The closed equation of system with no filter

$$G_{clNofilter} = \frac{5062500}{(s+8.995\times10^{4})(s+56.29)} \quad (5.20)$$

Figure 5-4 shows comparisons in the time and frequency domains.

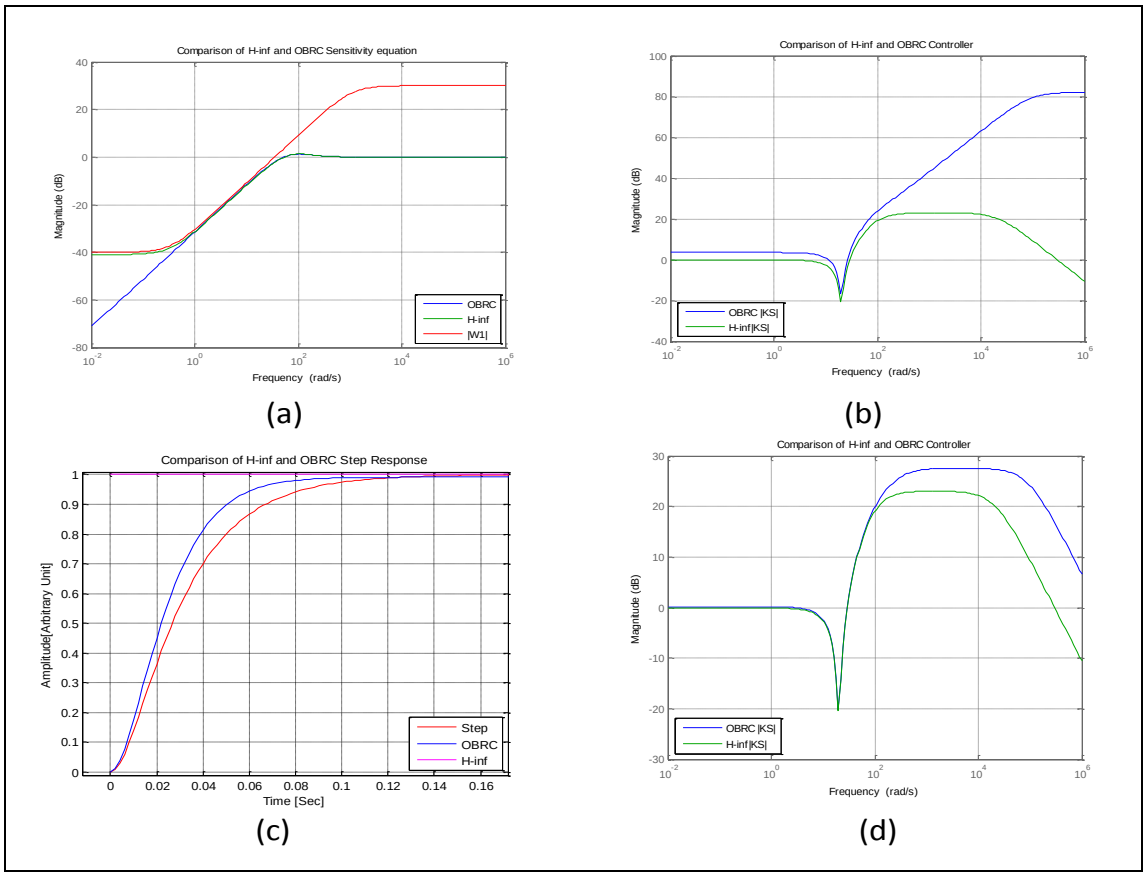


Figure 5-4: OBRC and H_∞ closed loop response

5.4 The RST Controller [13, 36, 27, 37]

The RST-controller initially was developed in the area of process industry and the designed methodology is based on z-domain. The acronym 'RST' was to identify this controller composed of the symbols chosen for its basic transfer function blocks as shown in figure 5.5. The polynomial controller of this subsection (5.5) has the same structure although it evolved independently. Hence the similarly defined acronym 'RHF' will be used to refer briefly to the polynomial controller. Since only pole placement in the Laplace domain is used to compare various control techniques in this thesis, the RHF controller will be considered instead of the RST controller henceforth'. The RST-controller name is derived from the structure of the controller as shown in Figure 5.5. The $T(z^{-1})$ is used as pre-compensator, $S(z^{-1})$ is filtering and $R(z^{-1})$ for the feedback derivative. The RST was initially developed in the area of process industry and the designed methodology is based on z-time domain and there are more than one approach for the pole placement techniques. The RST transfer function for the $T(z^{-1})$, $R(z^{-1})$ and $S(z^{-1})$ are

$$T(z^{-1}) = 1.419 - 4.358z^{-1} + 6.237z^{-2} + 5.011z^{-3} + 2.216z^{-4} - 0.4263z^{-5} \quad (5.21)$$

$$R(z^{-1}) = 1 + 0.3521z^{-1} \quad (5.22)$$

$$S(z^{-1}) = 1 + 0.3521z^{-1} \quad (5.23)$$

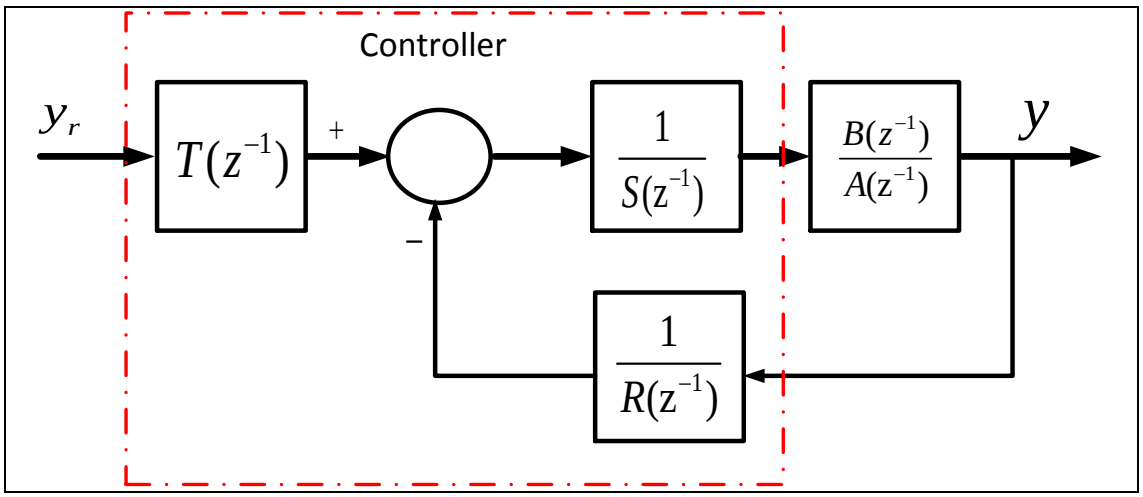


Figure 5-5: General block diagram of SISO RST-Controller

5.5 The Polynomial (RHF) Controller

5.5.1 Introduction

The design procedures for linear state feedback control systems are not straightforward and when employing observers rely on the separation principle which, in turn, demands an accurate plant model. Attempting pole placement with mismatched plant models is particularly onerous. In contrast, the technique of polynomial control achieves a similar performance to linear state feedback control for a given application through the process of complete pole assignment, but by using only the measured output without any additional need for the state feedback. Hence it eliminates the need for the observer in the design. Since, however, state estimation is often needed for nonlinear state feedback control and also the monitoring of estimated state variables is sometimes useful in linear control applications, observers still have a place in control engineering. Also state estimation can be regarded as an intrinsic process in polynomial control as it achieves complete control of the plant state and its own state!

A straightforward route to polynomial control is the realization that every linear continuous SISO controller is a linear system with two inputs $y_r(s)$, $y(s)$ and a single output $u(s)$ that can be represented by the transfer function relationship,

$$u(s) = G_r(s)y_r(s) - G_y(s)y(s) \tag{5.24}$$

This gives the control system structure shown in Figure 5.6.

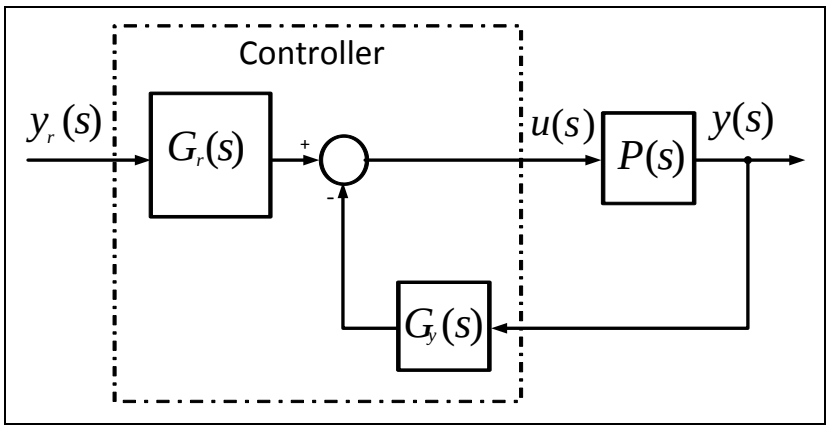


Figure 5-6: General linear feedback SISO block diagram

5.5.2 General Theory of the RHF Controller

The polynomial controller has a similar structure to the RST-controller as shown in Figure 5.7.

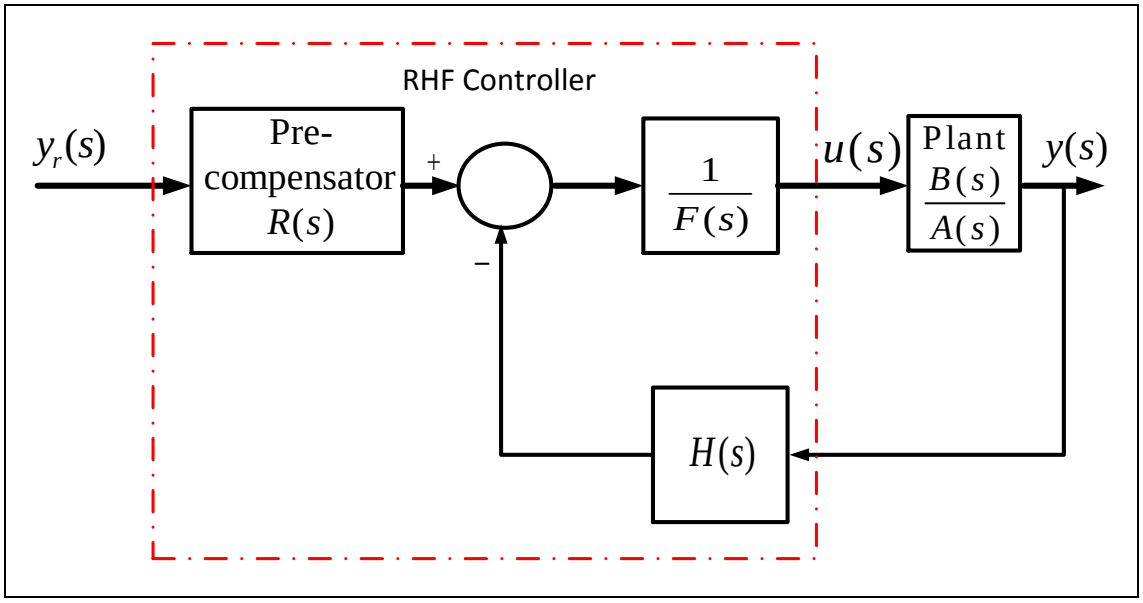


Figure 5-7: General block diagram of SISO RHF-Controller

The closed loop transfer function is

$$\frac{y(s)}{y_r(s)} = \frac{\frac{B(s)R(s)}{A(s)F(s)}}{1+\frac{B(s)H(s)}{A(s)F(s)}} = \frac{R(s)B(s)}{A(s)F(s)+B(s)H(s)} \tag{5.25}$$

This general expression will be useful in the design process to be developed later.

The transfer function relationship of the controller is

$$u(s) = \frac{R(s)y_r(s) - H(s)y(s)}{F(s)}\ , \tag{5.26}$$

where $R(s)$ is the reference input polynomial, $H(s)$ is the feedback polynomial, and the denominator polynomial, $F(s)$, will be called the filtering polynomial because equation (5.25) can be expressed as

$$u'(s) = R(s)y_r(s) - H(s)y(s), \qquad u(s) = \frac{1}{F(s)}u'(s) \tag{5.27}$$

in which any amplified high frequency components of measurement noise in $u'(s)$ due to the derivative action of $H(s)$ is attenuated before reaching the controller output $u(s)$, due to the low pass filtering action of the transfer function $\frac{1}{F(s)}$. This is equivalent to the filtering provided by an observer in a linear state feedback controller. It is clear that $R(s)$ is an external pre-compensator and the pre-compensator can be designed independently of the closed loop part. It also provides a means compensating the effect of the plant zeros without reducing the sensitivity if a further polynomial, $Z(s)$, is introduced as the denominator of the reference input transfer function, by replacing (5.26) with the following:

$$u(s) = \frac{1}{F(s)}\left[\frac{R(s)}{Z(s)} y_r(s) - H(s)y(s)\right] \tag{5.28}$$

The design procedure assumes the general plant transfer function is expressed in terms of its numerator and denominator polynomials, $A(s)$ and $B(s)$, which are respectively

$$A(s) = s^{na} + \sum_{i=0}^{n_a-1} a_i s^i \quad \text{and} \quad B(s) = \sum_{i=0}^{n_b-1} b_i s^i . \tag{5.29}$$

Also the relative degree of the open loop system has to be greater than zero to avoid an algebraic loop. Hence $n_a > n_b$. Later on an example of zero relative degree ($n_a = n_b$) will be used to present a procedure to handle this type of system.

Similarly, the controller polynomials are expressed as

$$\begin{cases} H(s) = \sum_{i=0}^{n_h-1} h_i s^i & F(s) = \sum_{i=0}^{n_f-1} f_i s^i \\ Z(s) = \sum_{i=0}^{n_z-1} z_i s^i & R(s) = \sum_{i=0}^{n_r-1} r_i s^i \end{cases} \tag{5.30}$$

The coefficients of the polynomials, $H(s)$ and $F(s)$, are the adjustable controller parameters used for the pole placement, the number of which can be minimised for given polynomial degrees. This allows the minimum order of closed loop system to be achieved for a given plant. The pre-compensator numerator polynomial is not

normalized in order to provide $n_r + 1$ coefficients, enabling its n_r zeros to be placed independently and the required closed loop DC gain to be set, which is usually unity. The functions of the polynomial controller terms are defined in Table 5.1.

Table 5.1: The RHF-controller functions

Components	*Purpose*
$H(s)$	To provide a sufficient number of adjustable coefficients in the controller to enable complete pole placement
$\frac{1}{F(s)}$	To prevent high frequency components of measurement noise amplified by $H(s)$ appearing as $u(s)$
$Z(s)$	To act as an external zero pre-compensator when needed, otherwise being set to unity
$R(s)$	To cancel some or all of the closed loop poles if desirable, otherwise $n_r = 0 \Rightarrow R(s) = r_0$ to become a simple reference input scaling coefficient

5.5.3 Constraints on Polynomial Degrees

With reference to the general closed loop transfer function (5.25), the closed loop characteristic polynomial is

$$A(s)F(s) + B(s)H(s). \tag{5.31}$$

Hence the order of the closed loop part of the system, excluding the pre-compensator, is

$$N = MAX\left[(n_a + n_f), (n_b + n_h)\right] \tag{5.32}$$

Importantly, the closed loop system cannot contain an algebraic loop. This is guaranteed if the relative degree of the open loop transfer function is positive. Thus,

$$\text{Reldeg}\left[\frac{B(s)H(s)}{A(s)F(s)}\right] = n_a + n_f - (n_b + n_h) > 0 \tag{5.33}$$

It then follows from (5.32) and (5.33) that the total system order is

$$N = n_a + n_f \tag{5.34}$$

It has already been stated that $F(s)$ is present to avoid high frequency measurement noise components amplified by the differentiating action being transmitted to $u(s)$, However, in order for this to be successful, the transfer function, $H(s)/F(s)$, must not exhibit any differentiating action revealed by the quotient of the polynomial division being a polynomial of degree 1 or greater. This transfer function must therefore have non-negative rank. Also, $n_h = n_a - 1$ to permit complete pole placement. Thus

$$n_f \geq n_h = n_a - 1 \tag{5.35}$$

5.5.4 Sensitivity

Since the pre-compensator in Figure 5-7 does not affect the sensitivity of the closed loop part, the simplified control loop of Figure 5-8 will be analysed.

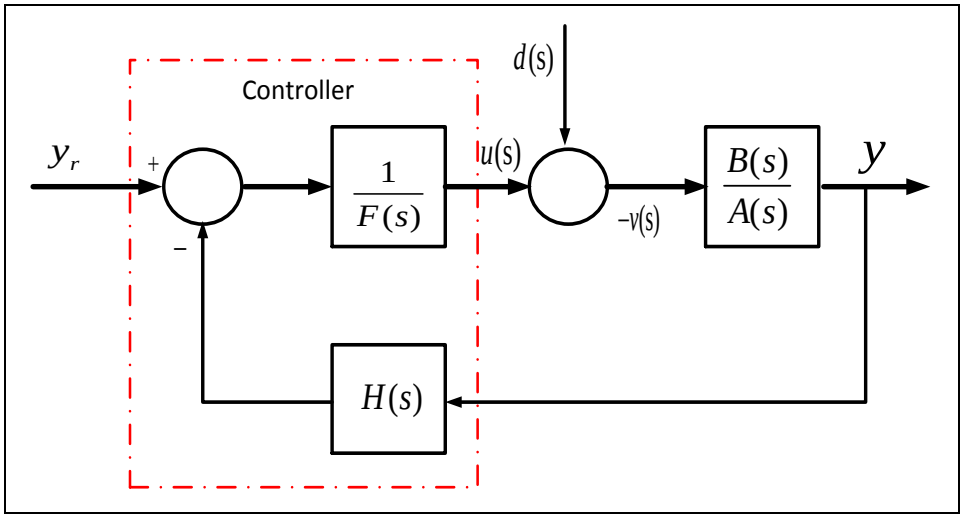


Figure 5-8 Simplified SISO block diagram of RHF-Controller for sensitivity analysis

The loop transfer function of this system is

$$L(s) = \frac{B(s)H(s)}{A(s)F(s)} \tag{5.36}$$

Using the standard notation for sensitivity analysis, the plant and controller transfer functions are, respectively, $P(s) = \frac{B(s)}{A(s)}$ and $C(s) = \frac{1}{F(s)}$. The sensitivity is then

$$S(s) = \frac{1}{1 + C(s)P(s)H(s)} = \frac{1}{1 + \dfrac{B(s)H(s)}{A(s)F(s)}} = \frac{A(s)F(s)}{A(s)F(s) + B(s)H(s)} \tag{5.37}$$

The same result can be obtained directly from Figure 5-8 as the transfer function between the net plant input, $v(s)$, and the disturbance input, $d(s)$. Thus

$$S(s) = \frac{v(s)}{d(s)} = \frac{1}{1 + \dfrac{B(s)H(s)}{A(s)F(s)}}$$

Hence, the sensitivity for the RHF-controller is given by

$$S(s) = \frac{A(s)F(s)}{A(s)F(s) + B(s)H(s)}. \tag{5.38}$$

5.5.5 Derivation of the General Pole Placement Equation

For pole placement of the closed loop system, the characteristic polynomial (5.31) will be set equal to a specified polynomial, $D(s)$, formed from the required closed loop pole locations. Thus

$$A(s)F(s) + B(s)H(s) = D(s). \tag{5.39}$$

A general algorithm for calculating the coefficients of $H(s)$ and $F(s)$ with a plant of arbitrary order will now be developed. The starting point will be the general linear second order plant with transfer function,

$$\frac{y(s)}{u(s)} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}. \tag{5.40}$$

The result obtained with this will enable the general pole placement equation to be deduced. The characteristic polynomial of (5.39) will be divided into two components, $A(s)F(s)$ and $B(s)H(s)$, as follows.

$$A(s)F(s) + B(s)H(s) = (s^2 + a_1 s + a_0)(f_2 s^2 + f_1 s + f_0) + k_i (h_2 s^2 + h_1 s + h_0)(b_1 s + b_0)$$

Consider first the $A(s)F(s)$ component,

$$\begin{aligned} A(s)F(s) &= (s^2 + a_1 s + a_0)(\, f_2 s^2 + \, f_1 s + \, f_0) \\ &= (s^4 + a_1 s^3 + a_0 s^2)\, f_2 + (s^3 + a_1 s^2 + a_0 s)\, f_1 + (s^2 + a_1 s + a_0)\, f_0 \end{aligned} \tag{5.41}$$

Equation (5.41) can be expressed in tabular form with the "f" coefficients and powers "s" as shown in Table 5.3.

Table 5.3: The coefficient of $F(s)$

	f_2	f_1	f_0
s^4	1	0	0
s^3	a_1	1	0
s^2	a_0	a_1	1
s^1	0	a_0	a_1
s^0	0	0	a_0

Similarly Table 5.4 can be formatted for the $B(s)H(s)$ term:

$$B(s)H(s) = (b_1 s^3 + b_0 s^{2)} h_2 + (b_1 s^2 + b_0 s) h_1 + (b_1 s + b_0) h_0 \tag{5.42}$$

Table 5.4: The coefficient of $H(s)$

	h_1	h_0
s^4	0	0
s^3	0	0
s^2	b_1	0
s^1	b_0	b_1
s^0	0	b_0

Combining Tables 5.3 and 5.4 yields Table 5.5 that contains the "f" and "h" coefficients.

Table 5.5: The Augmenting of the $F(s)$ and $H(s)$ Coefficients

	f_2	f_1	f_0	h_1	h_0
s^4	1	0	0	0	0
s^3	a_1	1	0	0	0
s^2	a_0	a_1	1	b_1	0
s^1	0	a_0	a_1	b_0	b_1
s^0	0	0	a_0	0	b_0

Then by observation of this table, the following linear matrix equation may be written.

$$\underbrace{\left[\begin{array}{ccc:cc} 1 & 0 & 0 & 0 & 0 \\ a1 & 1 & 0 & 0 & 0 \\ a_0 & a_1 & 1 & b_1 & 0 \\ 0 & a_0 & a_1 & b_0 & b_1 \\ 0 & 0 & a_0 & 0 & b_0 \end{array}\right]}_{M} \underbrace{\left[\begin{array}{c} f_2 \\ f_1 \\ f_0 \\ \cdots \\ h_1 \\ h_0 \end{array}\right]}_{K} = \underbrace{\left[\begin{array}{c} 1 \\ d_3 \\ d_2 \\ \cdots \\ d_1 \\ d_0 \end{array}\right]}_{D} \tag{5.43}$$

where ***M*** is the plant parameter matrix, **D** is the coefficient vector of the desired characteristic equation and **K** is the vector of controller parameters comprising the filter coefficients and feedback gains. Equation (5.53) is solved for **K.** Hence, later on the robustness controllers will be compared.

5.5.6 Basic RHF Controller Example

The design of the pre-compensator with reference input, $y_r(s)$ and output, $y_r'(s)$ is separate from the main RHF control loop and is considered in the following subsection. The closed loop system block diagram is shown in Figure 5.14 for plant (5.1).

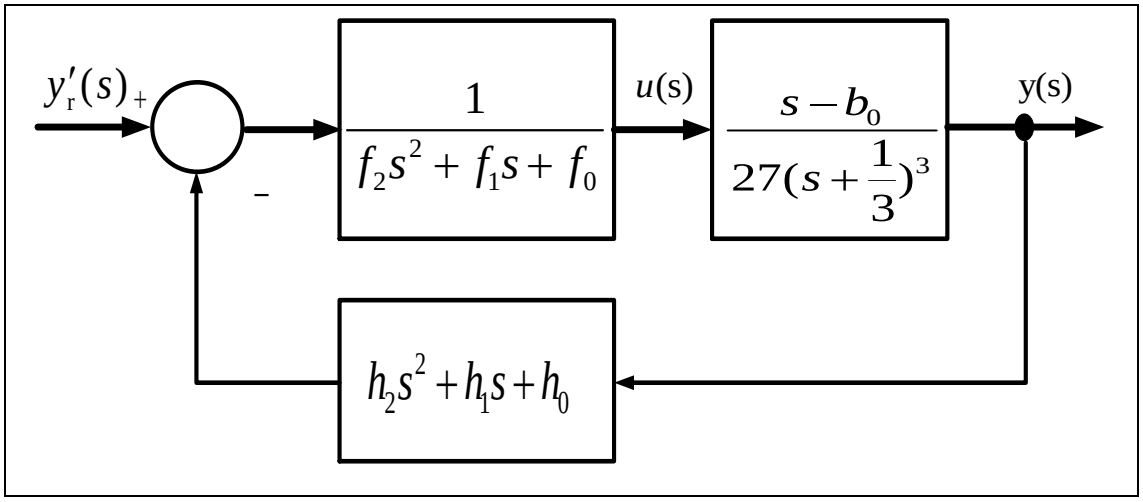


Figure 5-9: Closed loop system for RHF design

The general form of the characteristic equation obtained with this loop structure is $A(s)F(s) + H(s)B(s) = 0$. In this particular case it is

$$27(s + \frac{1}{3})^3 (f_2 s^2 + f_1 s + f_0) + (h_2 s^2 + h_1 s + h_0)(s - b_0) \tag{5.44}$$

To determine the controller parameters using the settling time formula (2.23), there are two options. The first is to place all five closed loop poles at one point. The second option is robust pole placement in which the closed loop poles are split into two groups, one dominant group at one location and the other at another location with a much larger magnitude to give additional robustness. With plant uncertainty, however, there is a trade-off between the robustness and stability. In robust pole placement, the desired characteristic equation is reformulated and is given by

$$D(s) = \overbrace{\underbrace{\left[s + \frac{1.5(n+1)}{T_s} \right]^n}_{\text{Dominant Mode}}}^{V(s)} \overbrace{\underbrace{\left[s + \frac{1.5(m+1)}{T_{sf}} \right]^m}_{\text{Fast mode for robustness}}}^{W(s)} \tag{5.45}$$

Where $T_{sf} << T_s$. In the particular case being considered, $n+m=5$ and we can vary the number of dominant poles over the range, $1 \le n < 4$. To form the linear matrix equation to be solved for the controller parameters in the form of (5.53), plant (5.1) is expressed in the standard form as follows.

$$\frac{y(s)}{u(s)} = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} = \frac{\frac{1}{27}s - \frac{1}{27}}{s^3 + s^2 + \frac{1}{3}s + \frac{1}{9}} \tag{5.46}$$

Hence $b_1 = \frac{1}{27}$, $b_0 = \frac{-1}{27}$, $a_2 = 1$, $a_1 = \frac{1}{3}$ and $a_0 = \frac{1}{9}$. The closed loop transfer function of the system of Figure 5-9 is

$$\frac{y(s)}{y_r'(s)} = \frac{b_1 s + b_0}{(s^3 + a_2 s^2 + a_1 s + a_0)(f_2 s^2 + f_1 s + f_0) + (b_1 s + b_0)(h_2 s^2 + h_1 s + h_0)}$$

i.e.,
$$\frac{y(s)}{y_r'(s)} = \frac{B(s)}{A(s)F(s) + B(s)H(s)} \tag{5.47}$$

where $F(s) = (f_2 s^2 + f_1 s + f_0)$, $H(s) = (h_2 s^2 + h_1 s + h_0)$, $A(s) = s^3 + a_2 s^2 + a_1 s + a_0$ and $B(s) = b_1 s + b_o$.

5.5.7 Example of RHF Controller with Integral Action

To eliminate steady state errors with a constant reference input or disturbance input, an integral outer loop may be added to the basic RHF control loop, as shown in Figure 5-10.

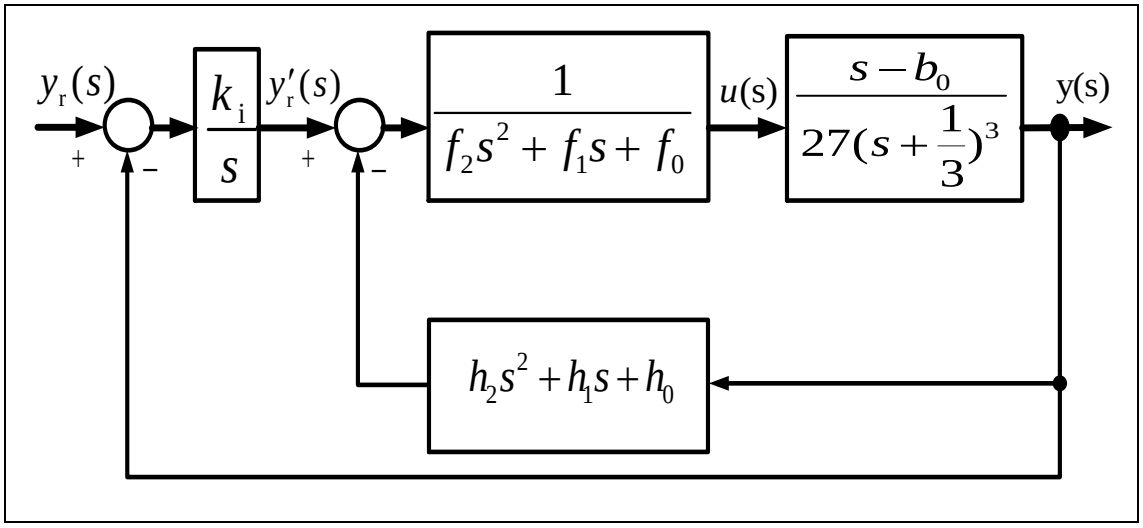


Figure 5-10: Closed loop system for RHF design with integral action

Then the closed loop transfer function becomes

$$\frac{y(s)}{y_r(s)} = \frac{B(s)k_i}{s\left[A(s)F(s) + B(s)H(s)\right] + B(s)k_i} \tag{5.48}$$

The characteristic polynomial of the closed loop system is

$$sA(s)F(s) + sB(s)H(s) + B(s)k_i \tag{5.49}$$

$$= s(s^3 + a_2s^2 + a_1s + a_0)(f_2s^2 + f_1s + f_0) + s(b_1s + b_0)(h_2s^2 + h_1s + h_0) + (b_1s + b_0)k_i$$

$$= (s^4 + a_2s^3 + a_1s^2 + a_0s)(f_2s^2 + f_1s + f_0) + (b_1s^2 + b_0s)(h_2s^2 + h_1s + h_0) + (b_1s + b_0)k_i$$

$$\left(s^4 + a_2s^3 + a_1s^2 + a_0s\right)\left(f_2s^2 + f_1s + f_0\right) + \left(b_1s^2 + b_0s\right)\left(h_2s^2 + h_1s + h_0\right) + \left(b_1s + b_0\right)k_i$$

$$\begin{aligned} &= f_2s^6 + \left(f_1 + a_2f_2\right)s^5 + \left(f_0 + a_2f_1 + a_1f_2 + b_1h_2\right)s^4 \\ &\quad + \left(a_2f_0 + a_1f_1 + a_0f_2 + b_1h_1 + b_0h_2\right)s^3 + \left(a_1f_0 + a_0f_1 + b_1h_0 + b_0h_1\right)s^2 \\ &\quad + \left(a_0f_0 + b_0h_0 + b_1k_i\right)s + b_0k_i \end{aligned} \tag{5.50}$$

The desired characteristic polynomial is

$$s^6 + d_5s^5 + d_4s^4 + d_3s^3 + d_2s^2 + d_1s + d_0 \tag{5.51}$$

Then equating the coefficients of like powers of s in (5.69) and (5.70) yields the following linear matrix equation to be solved for the controller parameters.

$$\underbrace{\left[\begin{array}{ccc:ccc:c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ a_1 & a_2 & 1 & b_1 & 0 & 0 & 0 \\ a_0 & a_1 & a_2 & b_0 & b_1 & 0 & 0 \\ 0 & a_0 & a_1 & 0 & b_0 & b_1 & 0 \\ 0 & 0 & a_0 & 0 & 0 & b_0 & b_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_0 \end{array}\right]}_{\mathbf{M}} \underbrace{\left[\begin{array}{c} f_2 \\ f_1 \\ f_0 \\ h_2 \\ h_1 \\ h_0 \\ k_i \end{array}\right]}_{\mathbf{K}} = \underbrace{\left[\begin{array}{c} 1 \\ d_5 \\ d_4 \\ d_3 \\ d_2 \\ d_1 \\ d_0 \end{array}\right]}_{\mathbf{D}} \tag{5.52}$$

Hence

$$\mathbf{MK} = \mathbf{D}$$

and the solution is simply

$$\mathbf{K} = \mathbf{M}^{-1}\mathbf{D} \tag{5.53}$$

5.5.8 Pre-compensator

The pre-compensator is not necessary in this case, because the poles of the closed loop system are chosen to reduce the effect of the zero to acceptable proportions during the transient. In subsections 5.9.5 and 5.9.6, however, performance comparisons are made with and without a pre-compensator to further reduce the effect of the zero. This pre-compensator consists of a 3^{rd} order low pass filter was used as a pre-compensator with a step response settling time of $T_{so} = 0.1[\text{s}]$, the filtering time constant being $T_{so}/6$.

5.5.9 Sine Wave Oscillator Example

This is another example that illustrates the capability of the RHF-controller. It was used in paper [37] and the original plant model was given in the *z*-domain as shown in Figure 5.3, an RST controller being applied. The initial step was to convert from the *z*-domain to the *s*-domain using Tustin's approximation, resulting in the plant transfer function,

$$\frac{y(s)}{u(s)} = \frac{0.7692s^2 - 61.54s + 923.1}{(s^2 + 10.77s)}. \tag{5.54}$$

This, however, has zero relative degree, implying that the RHF-controller cannot be used directly but this problem was solved by inserting a pure integrator in the plant input. Also the integrator will bring the benefits of a) acting as a low pass filter and attenuating high frequency components of measurement noise before they reach the physical plant input, and b) ensuring zero steady state error for any constant external disturbance components referred to the control input. Although this integrator is an addition to the controller, it is treated as part of the plant for the purpose of determining the controller parameters. So the task is to design an RHF controller for the augmented plant that has the transfer function,

$$\frac{y(s)}{u'(s)} = \frac{0.7692s^2 - 61.54s + 923.1}{(s^2 + 10.77s)s} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}, \tag{5.55}$$

and therefore a relative degree of 1, where the physical plant input is $u(s) = \frac{1}{s}u'(s)$.

According to (5.48) and (5.49), the minimum order the system can have is $N = n_a + n_f = 2n_a - 1 = 5$. So the matrix, $\mathbf{M}$, is of dimension $(N+1)\times(N+1) = 6\times 6$. Thus

$$\mathbf{M} = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & 1 & 0 & b_2 & 0 & 0 \\ a_1 & a_2 & 1 & b_1 & b_2 & 0 \\ a_0 & a_1 & a_2 & b_0 & b_1 & b_2 \\ 0 & a_0 & a_1 & 0 & b_0 & b_1 \\ 0 & 0 & a_0 & 0 & 0 & b_0 \end{array}\right] \tag{5.56}$$

where $a_2 = 10.77$, $a1 = 0$, $a0 = 0$, $b2 = 0.7692$, $b1 = -61.54$ and $b0 = 923.1$.

With reference to (5.64), the desired characteristic polynomial is

$$D(s) = \left[s + \frac{1.5(n+1)}{T_s} \right]^n \left[s + \frac{1.5(m+1)}{T_{sf}} \right]^m = s^5 + d_4 s^4 + d_3 s^3 + d_2 s^2 + d_1 s + d_0 \tag{5.57}$$

where $n + m = N = 5$ and $n > m$. The settling times chosen for the dominant and fast modes are, respectively, $T_s = 0.1$[s] and $T_{sf} = 0.001$[s], for robustness. Hence

$$D = \begin{bmatrix} 1 \\ d_4 \\ d_3 \\ d_2 \\ d_1 \\ d_0 \end{bmatrix} \tag{5.58}$$

The linear matrix equation to be solved for the controller parameters is then

$$\underbrace{\left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & 1 & 0 & b_2 & 0 & 0 \\ a_1 & a_2 & 1 & b_1 & b_2 & 0 \\ a_0 & a_1 & a_2 & b_0 & b_1 & b_2 \\ 0 & a_0 & a_1 & 0 & b_0 & b_1 \\ 0 & 0 & a_0 & 0 & 0 & b_0 \end{array}\right]}_{\mathbf{M}} \underbrace{\left[\begin{array}{c} f_2 \\ f_1 \\ f_0 \\ h_2 \\ h_1 \\ h_0 \end{array}\right]}_{\mathbf{k}} = \underbrace{\begin{bmatrix} 1 \\ d_4 \\ d_3 \\ d_2 \\ d_1 \\ d_0 \end{bmatrix}}_{\mathbf{d}} \tag{5.59}$$

The solution is therefore $\mathbf{k} = \mathbf{M}^{-1}\mathbf{d}$ and the required controller polynomials are

$$F(s) = f_2 s^2 + f_1 s + f_0 \tag{5.60}$$

with $f_2 = 1$ from (5.58), and

$$H(s) = h_2 s^2 + h_1 s + h_0 \tag{5.61}$$

5.6 The Internal Model Controller

In this section we will be comparing the performance of the RHF-Controller with IMC [21, 22, 29] and provide a brief overview of the IMC. For the purpose of the comparison will be using the plant with Right Hand Zero "RHZ" and the transfer function of the plant is shown by the equation 5.1.

5.6.1 Formulation of the IMC

The IMC structure was initially developed by Garcia and Morari in the early 1980's as shown in Figure 5.11.

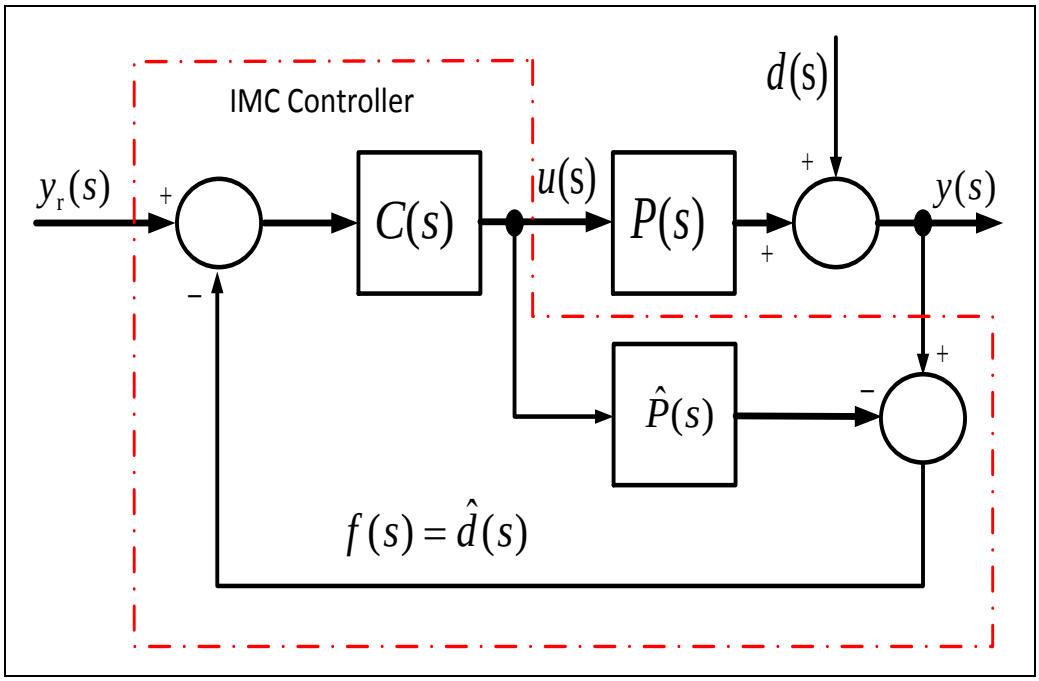


Figure 5-11: General SISO block diagram for IMC

The IMC concept makes use of the notion that the plant model is inaccurate and is a linearised model with transfer function, $\hat{P}(s)$. The real plant is modelled by a block with transfer function, $P(s)$, and a disturbance, $d(s)$, referred to the output, as shown. The disturbance, $d(s)$, is assumed unknown as it represents the plant modelling errors as well as unknown external disturbances. Thus

$$y(s) = P(s)u(s) + d(s)$$

The feedback signal in Figure 5_11 is given by

$$f(s) = P(s)u(s) + d(s) - \hat{P}(s)u(s) = \left[P(s) - \hat{P}(s) \right] u(s) + d(s)$$

Since $f(s) = d(s)$ if $\hat{P}(s) = P(s)$, then $f(s)$ may be regarded as a disturbance estimate, $\hat{d}(s)$. It is evident by inspection of Figure 5-11 that if $C(s) = \hat{P}^{-1}(s)$ and $\hat{P}(s) = P(s)$, then the feedback signal will precisely cancel the disturbance, $d(s)$. However in reality, the plant model is not a perfect match and therefore $d(s)$ will be a combination of the plant uncertainty and disturbance [3]. Nevertheless with $\hat{P}(s) \cong P(s)$, a degree of disturbance cancellation would be expected and therefore the control loop structure shown should have considerable robustness.

The closed loop transfer function of Figure 5.11 is

$$y(s) = \frac{C(s)P(s)y_r(s) + \left[1 - C(s)\hat{P}(s) \right] d(s)}{1 + \left[P(s) - \hat{P}(s) \right] C(s)} \tag{5.62}$$

It is evident that if $C(s) = \hat{P}^{-1}(s)$, and $P(s) = \hat{P}(s)$ then $\frac{y(s)}{y_{\mathrm{r}}(s)} = 1$, which implies perfect tracking of the set point and disturbance rejection. Although in practice, $\hat{P}(s) \neq P(s)$ since the plant model cannot perfectly match the actual plant, a perfect disturbance rejection can still be achieved by adjusting $C(s)$ until $C(s) = \hat{P}^{-1}(s)$. Then to achieve acceptable robustness of the closed loop system, we must ensure that the estimated model is closely matched to the real plant as possible. In general the model mismatch will be more pronounced at the higher frequencies. Hence, a low pass filter with transfer function, $C_f(s)$, is required in series with the controller, yielding a net controller transfer function of $C_{\mathrm{imc}}(s) = C(s)C_f(s)$. This also attenuates the noise caused by the derivative action of the controller, due to the inversion of the transfer function of the plant and mismatch of the model. Clearly in practice the plant is not always invertible and hence, the plant model can be divided into components, one which is invertible and another which is non-invertible, respectively $P^+(s)$ and $P^-(s)$, and then setting $C(s) = \left(P^+(s) \right)^{-1}$.

5.6.2 Sensitivity

For the purpose of sensitivity analysis, Figure 5-12 shows the unity feedback version of the IMC loop. It may be readily shown that its closed loop transfer function is the same as that of Figure 5-11.

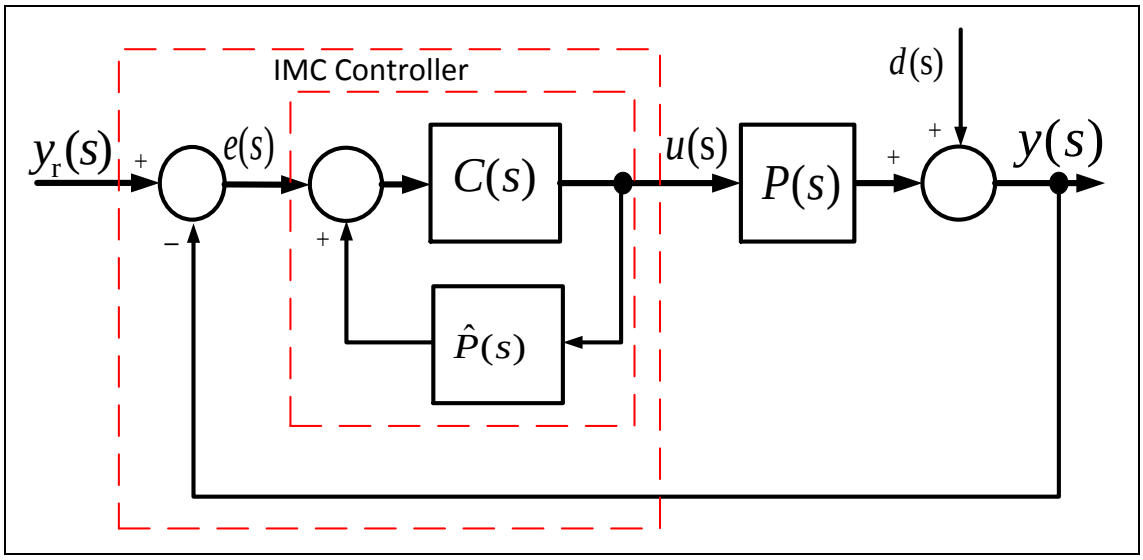


Figure 5-12: IMC block diagram with unity feedback

The sensitivity transfer function is

$$S(s)=\frac{y(s)}{d(s)}=\frac{1-C(s)\hat{P}(s)}{1-C(s)\left[\hat{P}(s)-P(s)\right]}$$

Remarkably, since $\hat{P}(s)$ is known precisely, then provided it is feasible to set $C(s)=\hat{P}^{-1}(s)$, then the sensitivity will be zero, implying ideal robustness. As pointed out in the previous subsection, however, this inversion is only possible for a factor of $P(s)$ in many cases.

5.6.3 Design Procedure

Step 1: Split the transfer function into two components “invertible and non-invertible”

Step 2: Determine the open loop response of the plant.

Step 3: Design the controller transfer function as shown in equation 5.63

$$C_{\text{imc}}(s) = \frac{\left(P^{+}(s)\right)^{-1}}{(1+\tau_f s)^n} \tag{5.63}$$

where *n* is the order of the filter and τ_f is the time constant of filter. The bandwidth of the filter needs to be twice the bandwidth of the open loop plant.

The IMC controller was designed using the Matlab SISO Tool and the controller is

$$C_{\text{imc}}(s) = \frac{-19.3205(s+0.333)^3}{s(s^2+2.709s+3.17)} \tag{5.64}$$

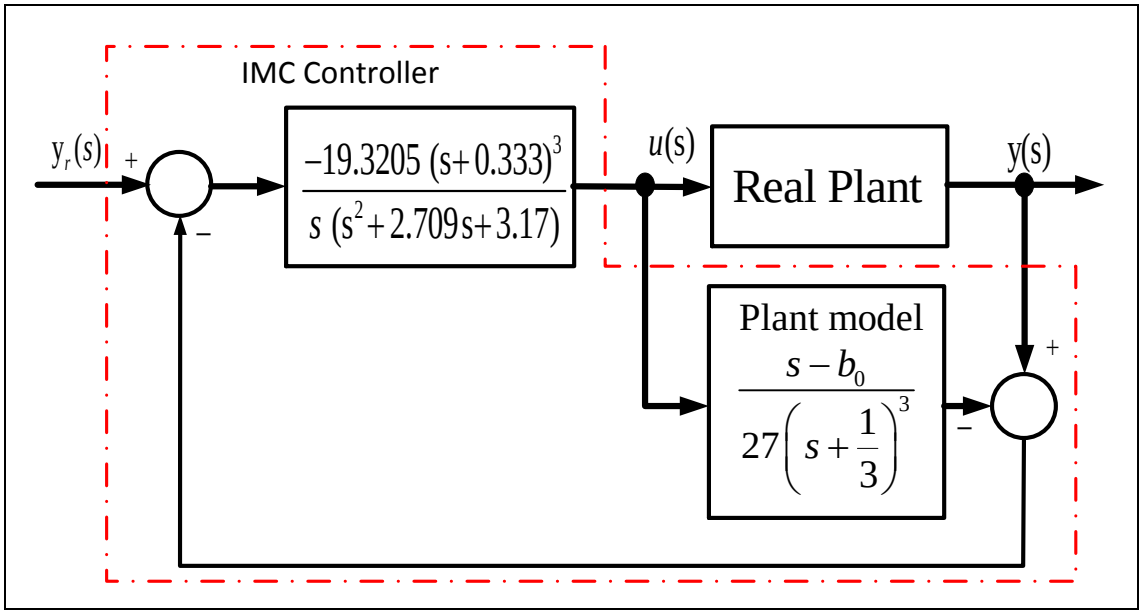


Figure 5-13 Block diagram of SISO IMC system

5.7 The H-Infinity Controller

5.7.1 Overview

The H-infinity control theory is well documented in [12, 45] and some further reading material can also be found in [45, 51, 41]. Hence this section concentrates on its application. The H-infinity methodology is based on a complex optimisation procedure and one must use an off-the-shelf software package to design the controller. The designer has to formulate a constraint based on the desired performance.
In this section, a design for plant (5.2) will be produced.

5.7.2 Formulation of the H-infinity Controller

The structure of the H-infinity control loop is the simple classical unity feedback one, as shown in Figure 5-14.

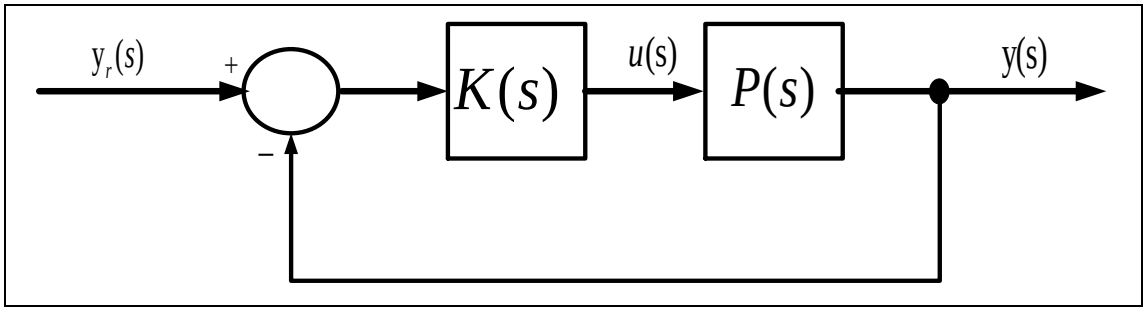


Figure 5-14: H-infinity Block diagram without constraints

The sophistication is in the determination of the controller transfer function, $K(s)$.
The key steps for the designer are to understand the plant, beware of the tool pitfalls, and be able to formulate the design specification into the weighting function which is then used as a constraint in the software tool. Table 5.6 presents a specification, and the terms of which are explained in the following subsection.

Table 5.6: Typical design specification

Controlparameter	*Requirement*
ω_b	The closed loop bandwidth must not excess 35.6 [rad/s]
$\|K_S\|$	The gain must not be greater than 5.5
$\|Offset\|$	The steady state error must not be greater than 0.01
γ	The optimization must be less than 1.33

5.7.3 Design Specification

In Table 5.6, γ is the minimisation criteria and the constraints are formed by taking the transfer function between the disturbance input, $d(s)$ and outputs $Z_1(s), Z_2(s), \ldots$.

Hence, the minimum constraints for three outputs are

$$\min \gamma \left\| \begin{matrix} W_1.S \\ W_2.K.S \\ W_3.T \end{matrix} \right\| < \gamma \qquad (5.65)$$

where

W_1 is the weighting function associated with sensitivity function,

W_2 is the weighting function associated with sensitivity function and control penalising high frequency components, and

W_3 is the weighting function associated with complimentary sensitivity function.

The general block diagram of Figure 5.15 shows the outputs for forming the constraints for this problem, which will be used by the H-infinity process to design the controller, $K(s)$, that meets the design specification and the general form for designing the $W_1(s)$ and $W_2(s)$. This design specification is given by equation (5.58) and the constraints plotted on the magnitude as $\frac{1}{W}$.

$$W_1(s) = \frac{s/M + \omega_B}{s + \omega_B A}, \quad W_3(s) = \frac{s + \omega B/M}{As + \omega_B} \qquad (5.66)$$

where:

M is magnitude of the controller.

A is the amplitude of the error.

ω_B is the closed loop band width of the plant.

Once the design specification is translated into constraints, then the rest of it is handled by the tool that designs the controller. The plant transfer function (5.2) is repeated here, as follows.

$$P(s) = \frac{b}{s^2 + a_1 s + a_0} \tag{5.67}$$

where $b = 400$, $a_1 = 2$ and $a_0 = 400$

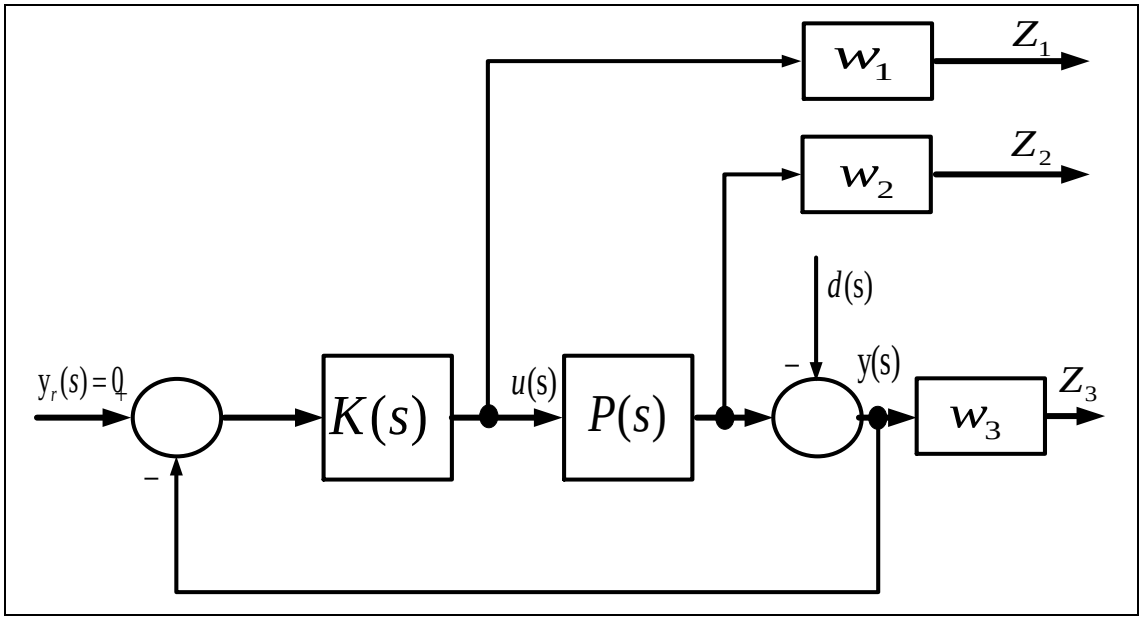


Figure 5-15: H-infinity Block diagram showing constraint outputs

5.7.4 Steps to use the Matlab Toolbox

```
s = tf('s');
G = 400/(s^2 + 2* s +400);
Gss = ss(G);
ssg = mksys(Gss.a,Gss.b,Gss.c,Gss.d);

W1 = (0.03162*s+35)/(s+0.35); %
W2 = 0.06667;
W3 = (s+19.61)/ (0.01*s+62);
tss = augtf(ssg,W1,W2,W3);
[rhoopt,ssf,sscl] = hinfopt(tss);
gamma = 1/rhoopt;
[Ka,Kb,Kc,Kd] = branch (ssf);
K = tf(ss(Ka,Kb,Kc,Kd));
```

The controller $K(s)$ is produced by the optimization routine. Hence

$$K(s) = \frac{2.977\times10^{5}s^{3} + 1.847\times10^{9}s^{2} + 3.811\times10^{9}s + 7.384\times10^{11}}{s^{4} + 2.7929\times10^{4}s^{3} + 1.338\times10^{8}s^{2} + 1.904\times10^{10}s + 6.649\times10^{9}} \quad (5.68)$$

The loop transfer function is $L(s) = K(s)G(s)$. Hence the sensitivity function is $S = \frac{1}{(1+L)}$. Therefore

$$S = \frac{s^{4} + 2.729\times10^{4}s^{3} + 1.338\times10^{8}s^{2} + 1.904\times10^{10}s + 6.649\times10^{9}}{s^{4} + 2.729\times10^{4}s^{3} + 1.338\times10^{8}s^{2} + 1.916\times10^{10}s + 7.45\times10^{11}} \quad (5.69)$$

The complementary sensitivity function is defined as $T = 1 - s$. Hence

$$T = \frac{1.191\times10^{8}s + 7.384\times10^{11}}{s^{4} + 2.729\times10^{4}s^{3} + 1.338\times10^{8}s^{2} + 1.916\times10^{10}s + 7.45\times10^{11}} \quad (5.70)$$

5.8 Comparisons

5.8.1 Overview

In this section we will be presenting all the simulation results for the comparison. The controllers were subjected to parameter uncertainty, as well as some nonlinearities such as actuator (plant input) saturation. Although a numerous set of plots were collected, only a relevant selection of these is used for the comparison. Table 5.7 provides the list of controllers that were compared with the OBRC and RHF controllers.

Table 5.7: Control Methodology Comparison

Control Method		Control Method
H_∞	V	OBRC
IMC	V	RHF
RTS	V	RHF

5.8.2 Comparison of OBRC and H-infinity

For this comparison two variants of the simulation model were tested, the first with no saturation block at the input of the plant, as shown in Figure 5.16 and the second with a saturation limit on the input to plant, as shown in Figure 5.17. The reason for this comparison is due to the fact that in the real plant, the actuators and the fuel is limited based on the engine conditions.

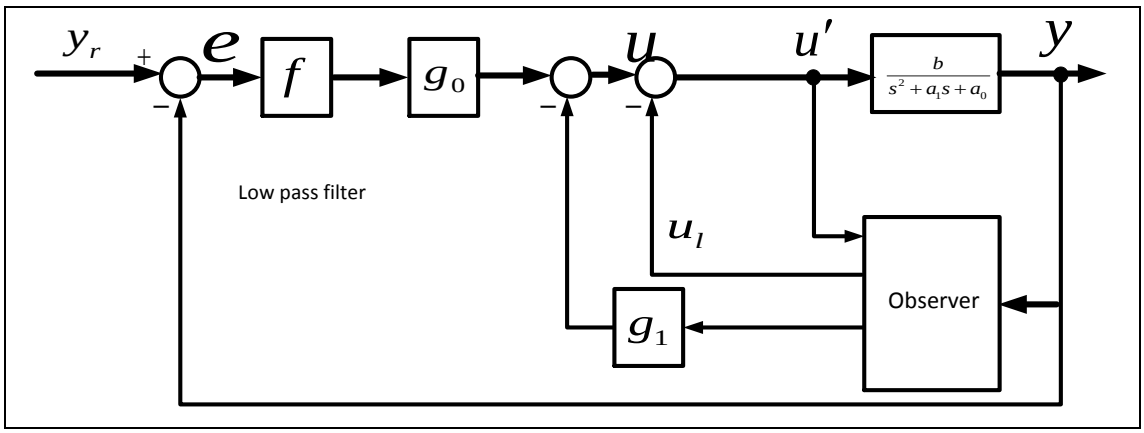


Figure 5-16: OBRC with low pass filtering in the error path without control saturation

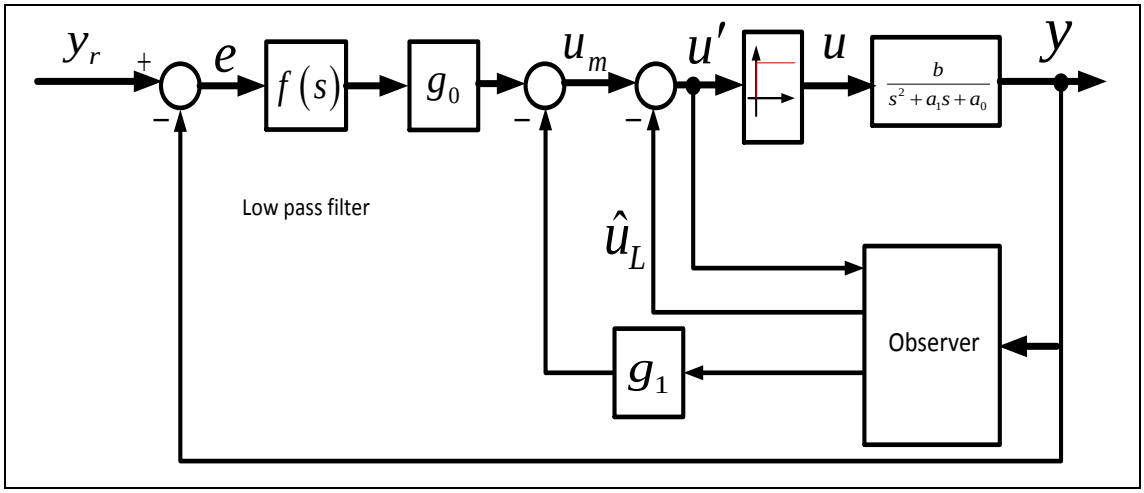


Figure 5-17: OBRC with low pass filtering in the error path with control saturation

Figure 5-18 shows the frequency responses for the low pass filter parameters used.

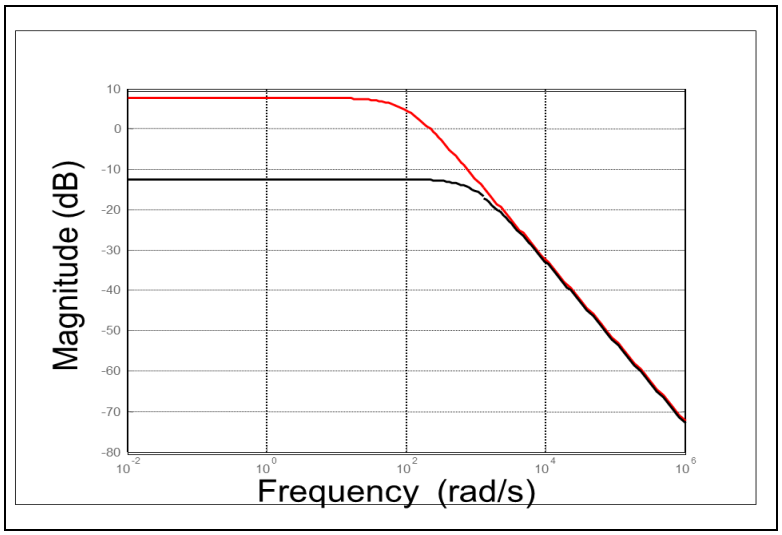


Figure 5-18: Frequency plot of filters with different time constants

Table 5.8 shows the parameters selected for the H_∞ controller.

Table 5.8: Parameters k_ρ and ω_n for the H_∞ controller

Test	k_ρ	ω_n
1	400	20
2	300	20

3	400	15
4	300	15

5.8.3 Comparisons with Nominal Plant Model

The intention in this simulation is to assess the performance of the controllers with the nominal plant model. This is done by comparing the step responses and the frequency responses. The effect of filtering can be seen on the error signal as shown in Figures 5.19 and 5.20.

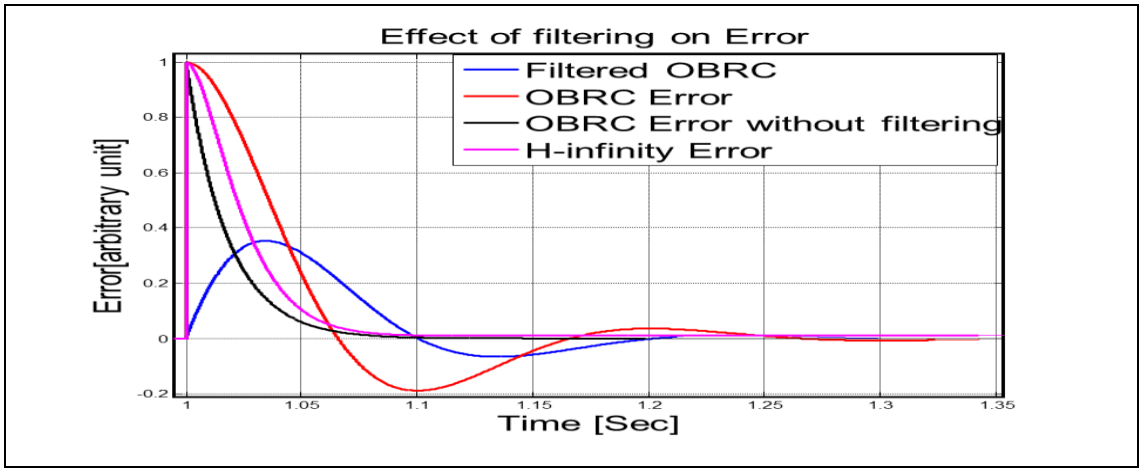


Figure 5-19: Plot of the with lower filter gain

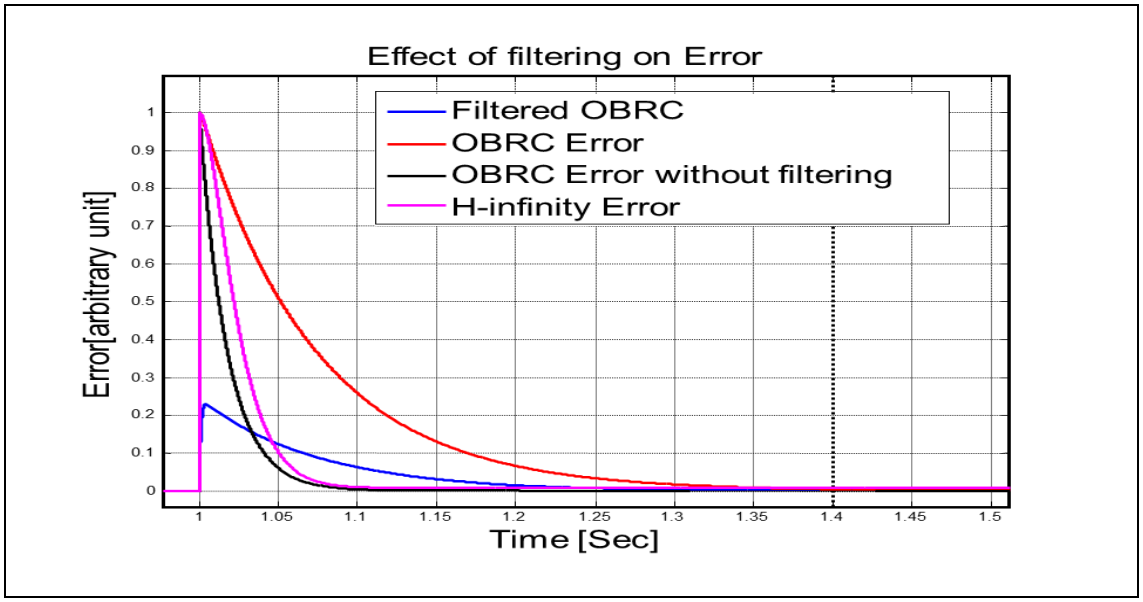


Figure 5-20: Effect of the higher gain filter on the error signal

Hence, it affects the controller's ability to see the plant output. The filter has several effects on the error. Its benefit is filtering the measurement noise but it effectively reduces the controller gains, the effect of which is to increase the settling time by a small amount as shown.

Figure 5.21 (a) is the step response of the OBRC and H-infinity and effect of the fast time constant and figure 5.21 (c) and (d) is the controller effort.

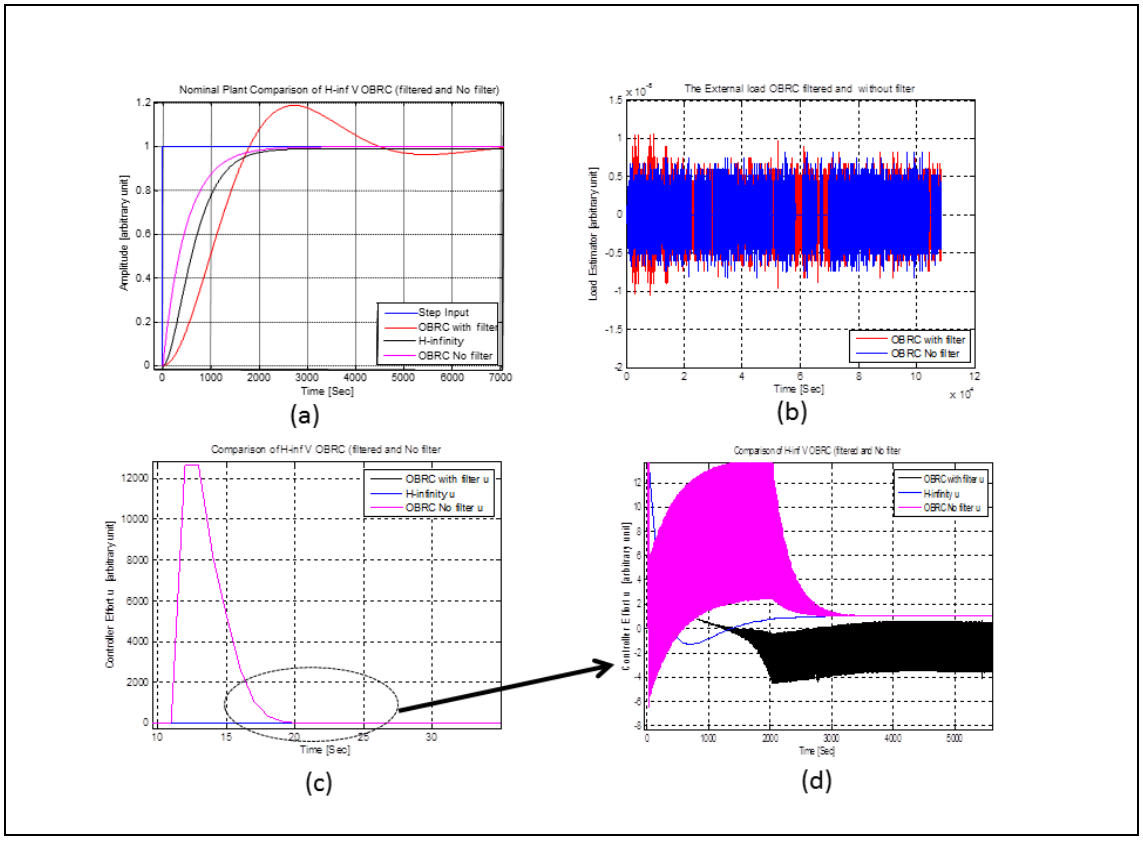


Figure 5-21: OBRC and H-infinity step response with Controller effect

We can see there is a high level of variation in the OBRC controller effort whereas the H-infinity has virtually no variation at all. We can see that Figure 5.21 (b) shows that the disturbance estimate is almost zero and this is expected for the nominal model. Figure 5.21(a) is the plot of the step response of the three controllers, which shows that the step response of the OBRC with no filtering is similar to the H_∞ design. However, the OBRC with filtering has a slight overshoot. The overshoot is due to the filtering of the error and therefore it restricts ability of the OBRC controller to control the plant. A possible way to overcome this limitation would be to use the estimate, $\hat{y}$, of y from the observer, that would attenuate the measurement noise through its filtering action and insert the low pass filter outside the loop, in only the reference input path, which would limit the control activity without introducing uncertainties in the closed loop dynamics that causes the under-damped response of Figure 5-21(a).

Figure 5.22(a) shows the step response of the two controllers. We can see that the H-infinity controller can no longer follow the reference signal as the system becomes unstable, whereas the OBRC controller follows the reference signal. In figure 5.22(b) we see the change as a disturbance or mismatch.

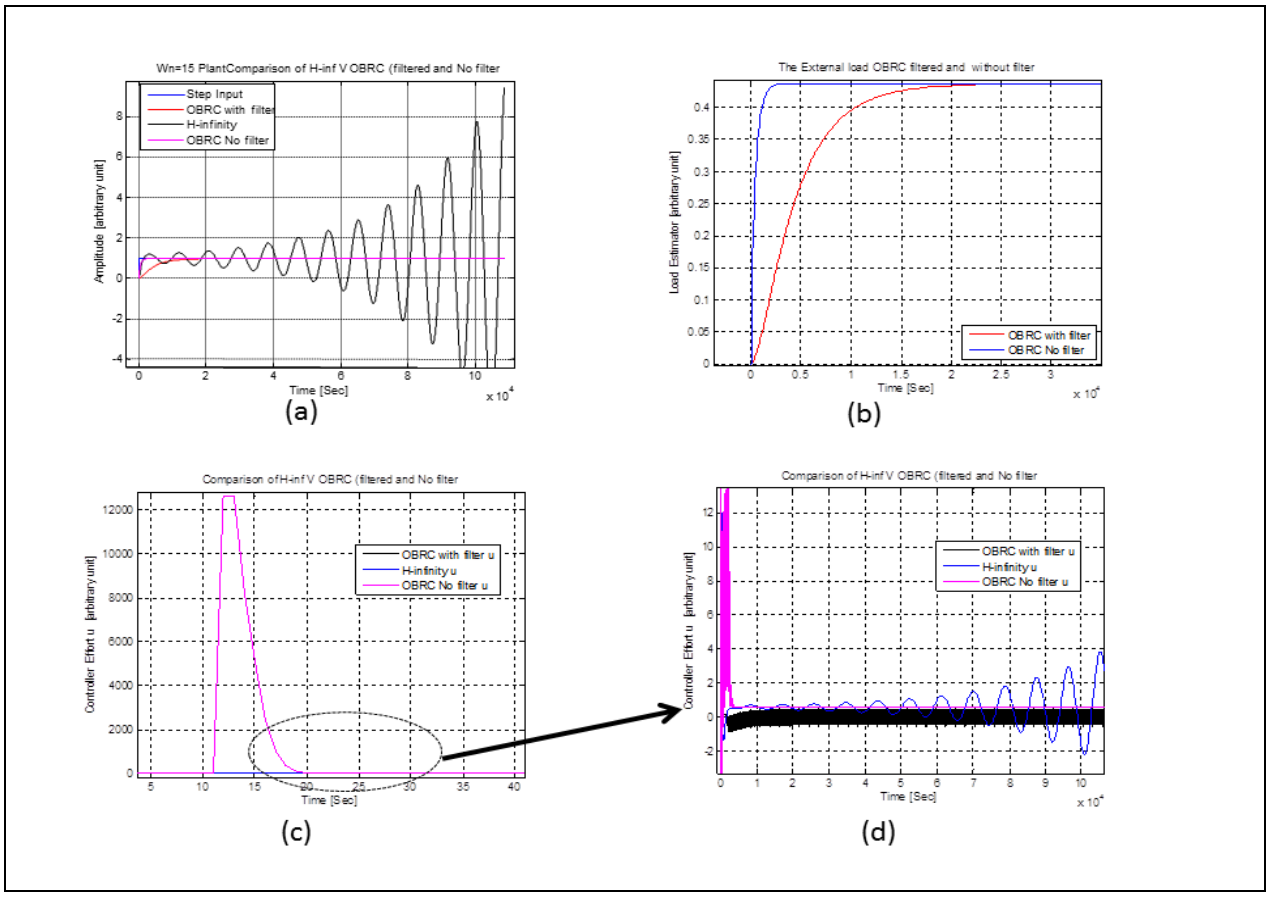
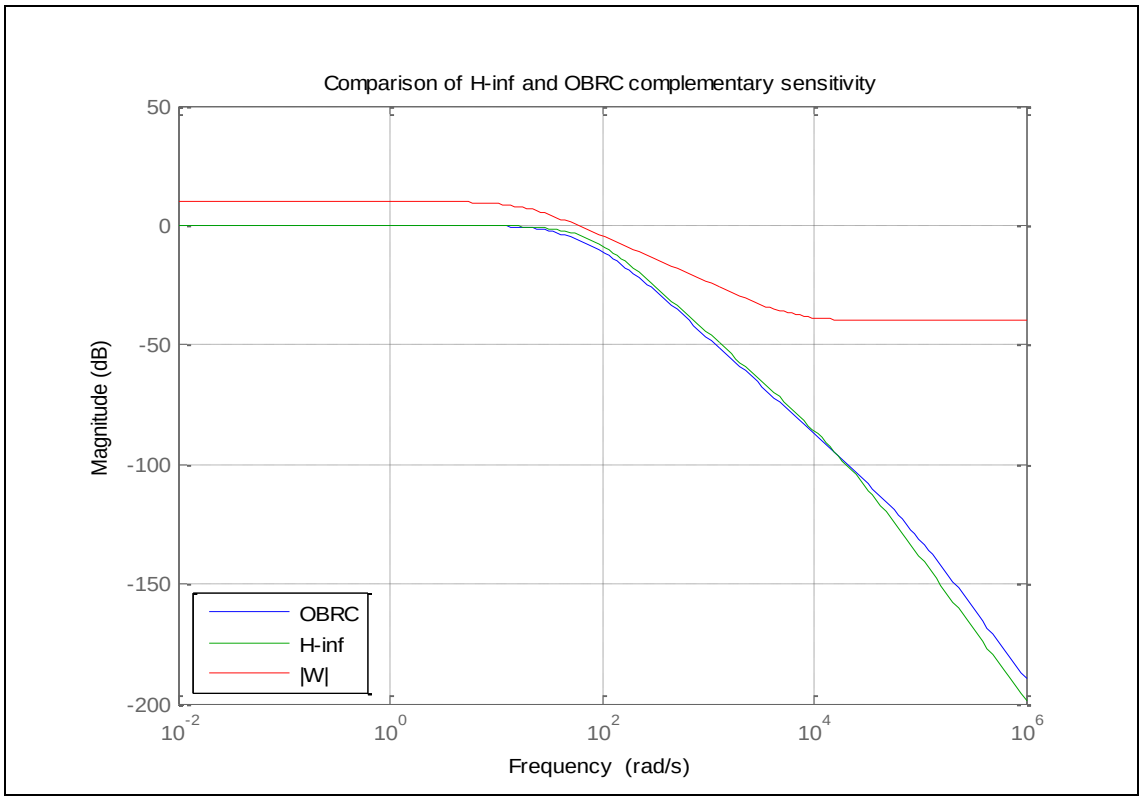


Figure 5-22: Step Response to change $\omega_n = 15$ (25% reduction)

Figure 5.22 shows the effect of the saturation on the controller's to a step response and we can see the OBRC controller with slow filter time constant has managed to follow the reference input and not overshoot. The penalty for this was an increase in the settling time. We can see from Figure 5.19(b) that the external disturbance is measuring this as mismatch, and also that the error filtering is filtering the effect of the mismatch. Figure 5.19(d) shows the controller activity in the steady state.

The OBRC and H-infinity controllers have nearly the same frequency response as shown in Figure 5.23.



igure 5-23: Frequency plot of the closed loop system for OBRC and H-infinity

The frequency response plot of the $|K(s)S|$ given in Figure 5.24, however, shows that at high frequencies the OBRC controller will amplify the measurement noise.

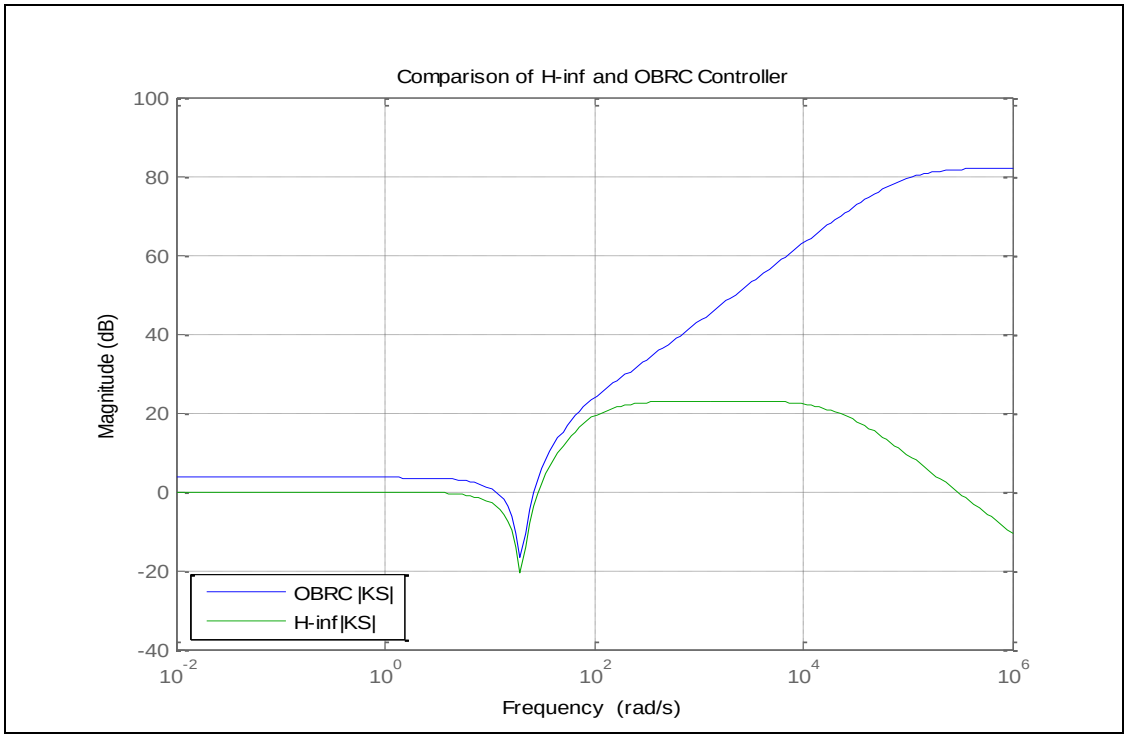


Figure 5-24: frequency plot of the $|KS|$ OBRC with no error filtering and H-infinity

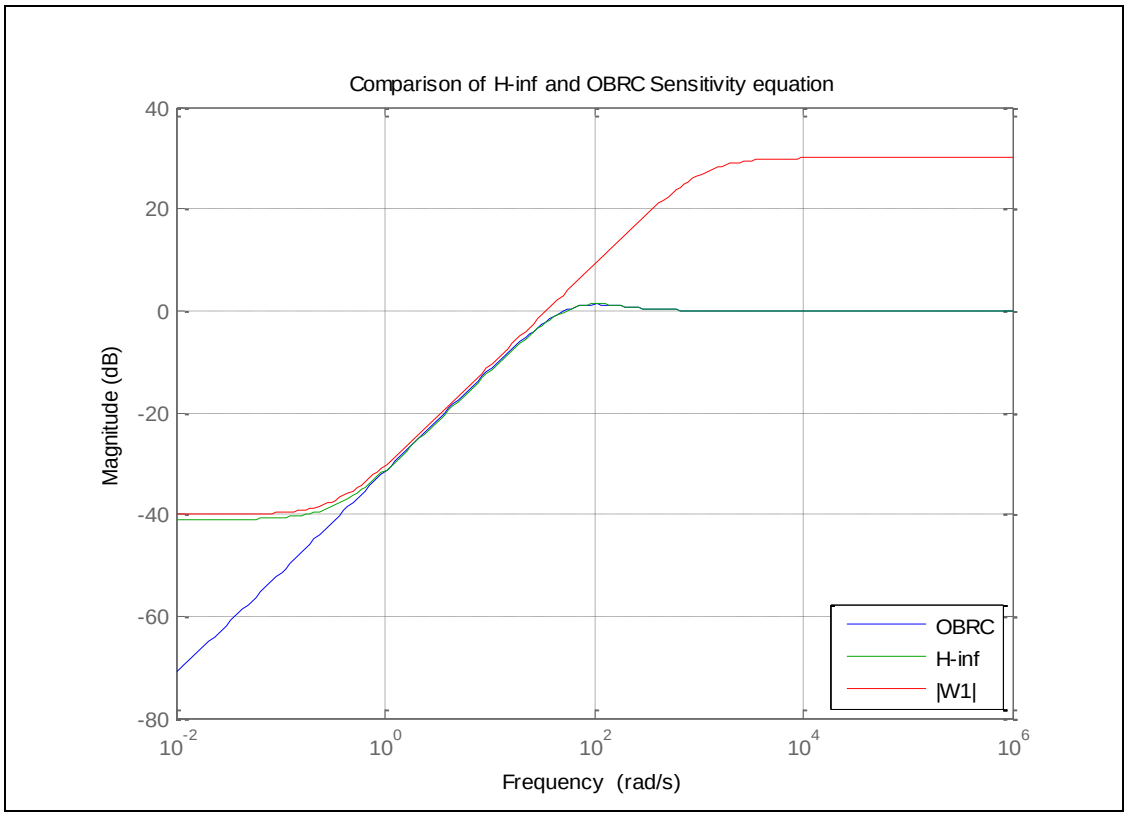


Figure 5-25: frequency plot of the $|S|$ and $|1/W_1|$ for OBRC and H-infinity

The closed loop step response of the OBRC is not effected by introduction of the filter as shown in Figure 5.26 and 5.28.

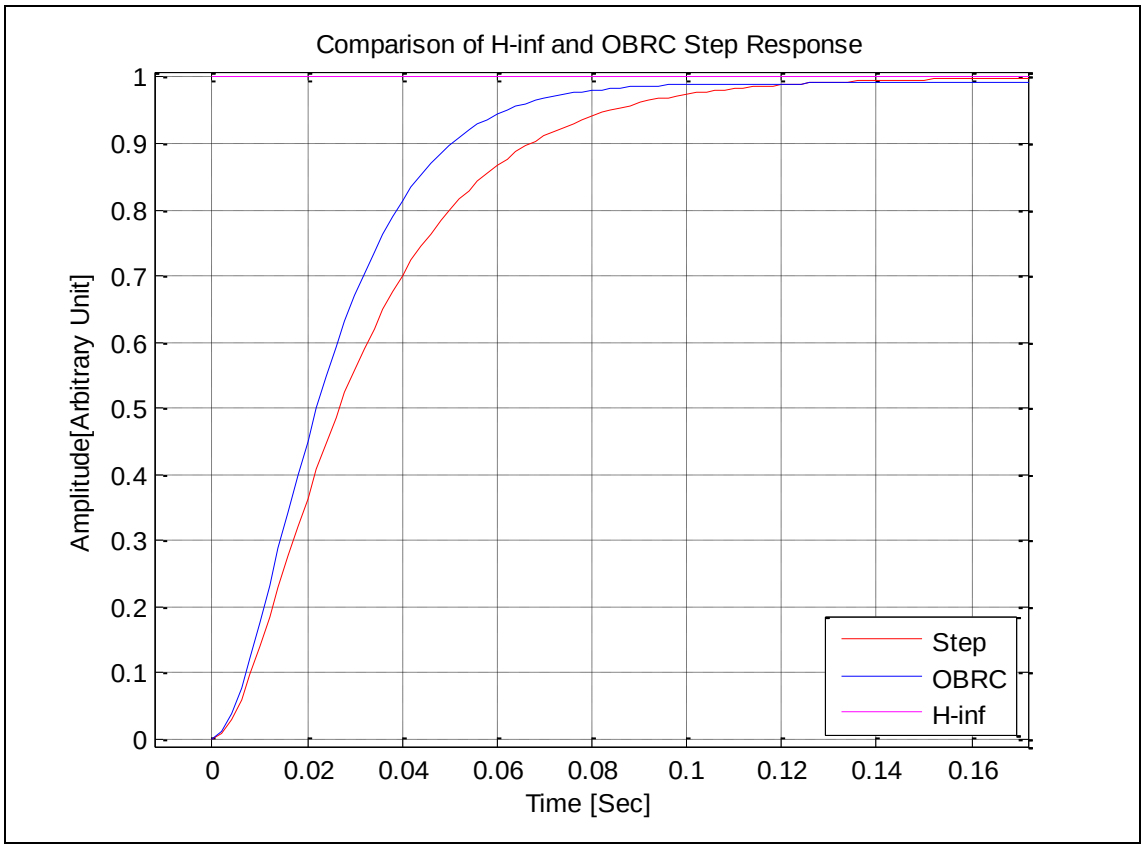


Figure 5-26: Step response of the OBRC with no error filtering and H-infinity

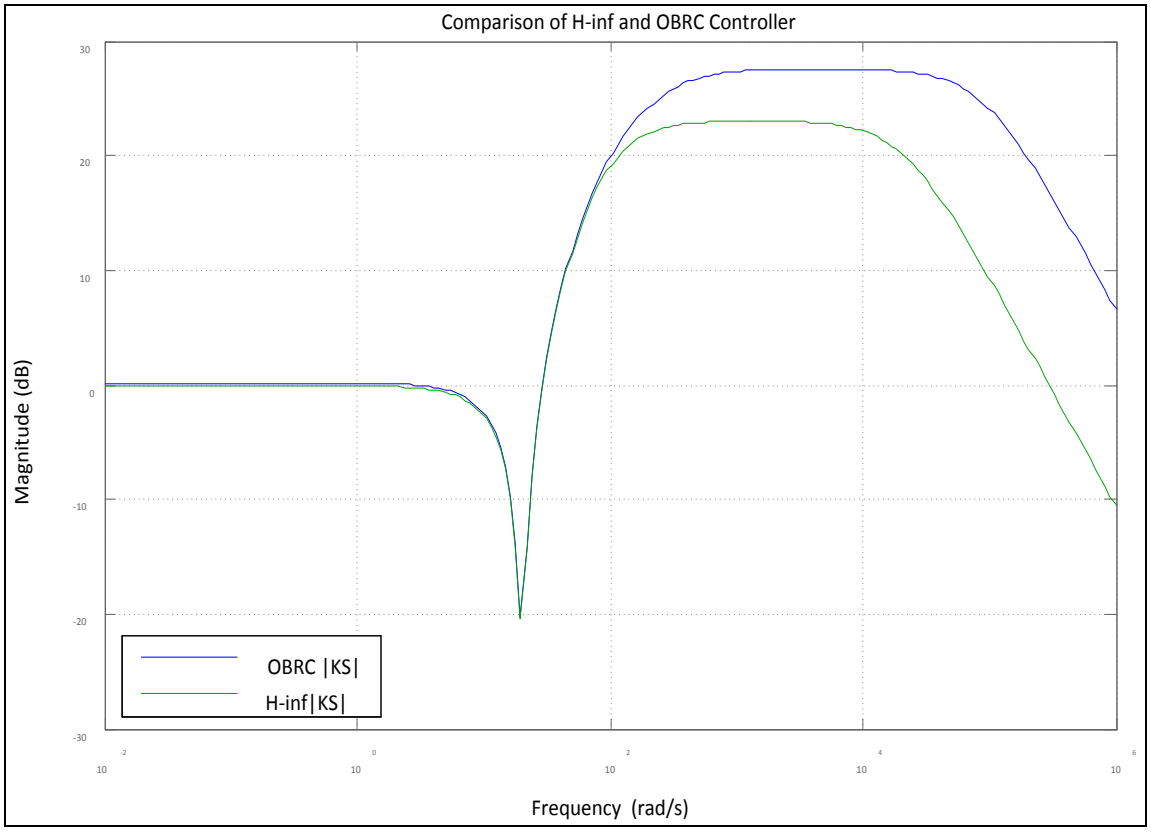


Figure 5-27: Plot of the $|KS|$ OBRC with error filtering and H-infinity

Then, we can see the effect of error filtering in Figure 5.27 which minimises the controller effort at the higher frequency. In general the frequency responses of both controllers are similar, except that the H-infinity has a higher order closed loop system.

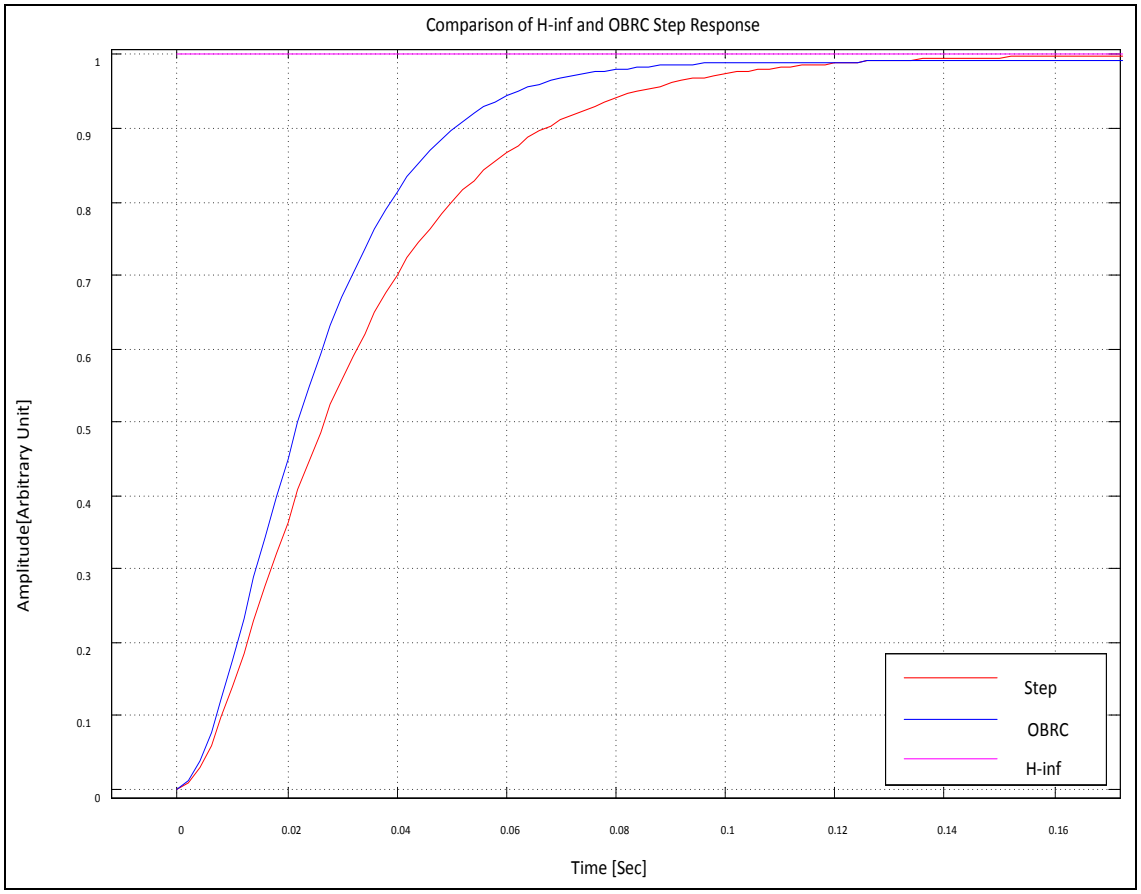


Figure 5-28: Step response of the OBRC with error filtering and H-infinity

Figure 5.29 shows the sensitivity frequency response $|S(s)|, s = j\omega$, to a 25% change in ω_n parameter, which has affected the $|S(s)|$ of the H_∞ frequency response near the bandwidth of the plant, whereas no change in the frequency response of the OBRC $|S(s)|$.

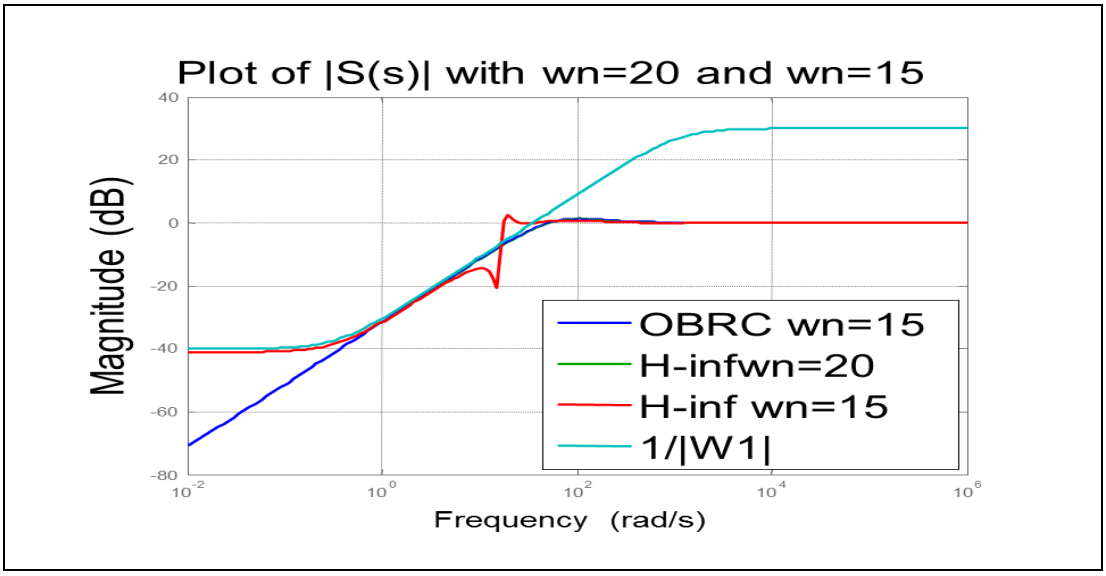


Figure 5-29: Effects on S(s) when saturation is introduced at the Plant input

The change in ω_n has been reflected in the $|SK(s)|$ of both OBRC and H_∞ controllers, as shown in Figure 5.30. The change in ω_n has been seen by the OBRC controller in the higher frequency region, whereas the H_∞ controller is effected in the lower frequency region. Hence, it has reduced the robustness of the H_∞ controller.

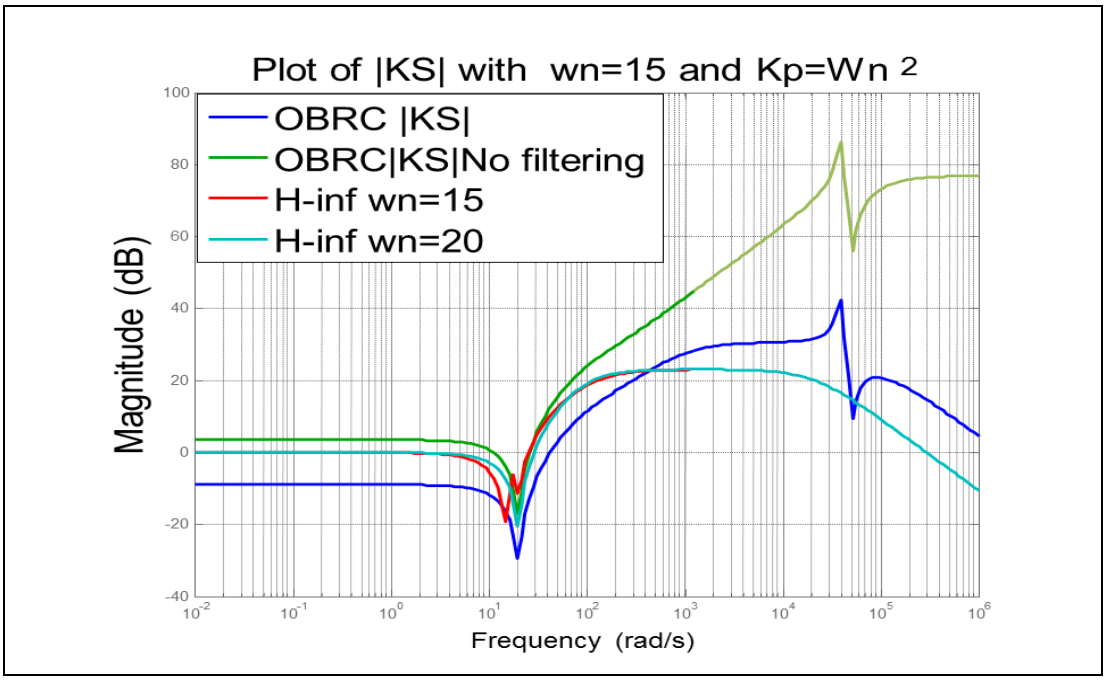


Figure 5-30: Further effects on |KS(s)| when saturation is introduced at the plant input

5.8.4 Comparisons with Parameter Variations

The plant parameters were varied as shown in Figure 5-31, without changing the controller parameters. The reason for this type of simulation is to assess the robustness against plant parameter variations and model uncertainties.

The list of Parameter Values

Test Number	a2	a1	a0	b0	k
1	1	0.3333	0.0370	- 1	0.0370
2	0.99778	0.31544	0.037194	-0.87594	0.038424
3	1.097596	0.324496	0.036263	-1.46226	0.042345
4	0.919742	0.353079	0.040098	-1.23813	0.048449
5	1.044245	0.309104	0.038368	-1.39324	0.032159
6	1.043007	0.351937	0.036994	-0.59628	0.041593
7	0.939562	0.346583	0.035809	-1.46946	0.04862
8	1.080944	0.331995	0.037037	-0.89013	0.044269
9	1.015344	0.353699	0.0397	-1.31708	0.040523
10	0.99798	0.301912	0.0399	-1.33207	0.029331
11	0.994218	0.333365	0.038613	-1.44038	0.05122

Figure 5-31: Plant parameter variations for robustness assessment

5.8.5 Comparison of RHF and IMC Controllers

Both controllers were simulated and the Matlab robust tool was used to vary the plant parameters to provide a measure of robustness against the model uncertainty. In total 11 simulation runs were performed, with the coefficients of the plant being varied as shown in the Figure 5.31 and the first test run being done with the nominal plant parameters. The

degrees of the factors, $V(s)$ and $W(s)$, of the desired characteristic polynomial, $D(s) = V(s)W(s)$, of the RHF controller [ref., equation (5.64)] were varied to see the effect on the closed loop performance for the same settling time. A small selection of the many simulation results was made for the assessment, but the complete set is provided in Appendix I. When the degree of $\deg[V(s)] \leq \deg[W(s)]$ then the closed loop system was not sufficiently robust and even became unstable. A better closed loop performance was found when the order of $\deg[V(s)] > \deg[W(s)]$.

The Bode plots shown in Figure 5.32 span a wide range of plant parameter variations (typically up to $\pm 30\%$) used in testing the controller performance against the model uncertainty. The controllers were designed for the nominal plant.

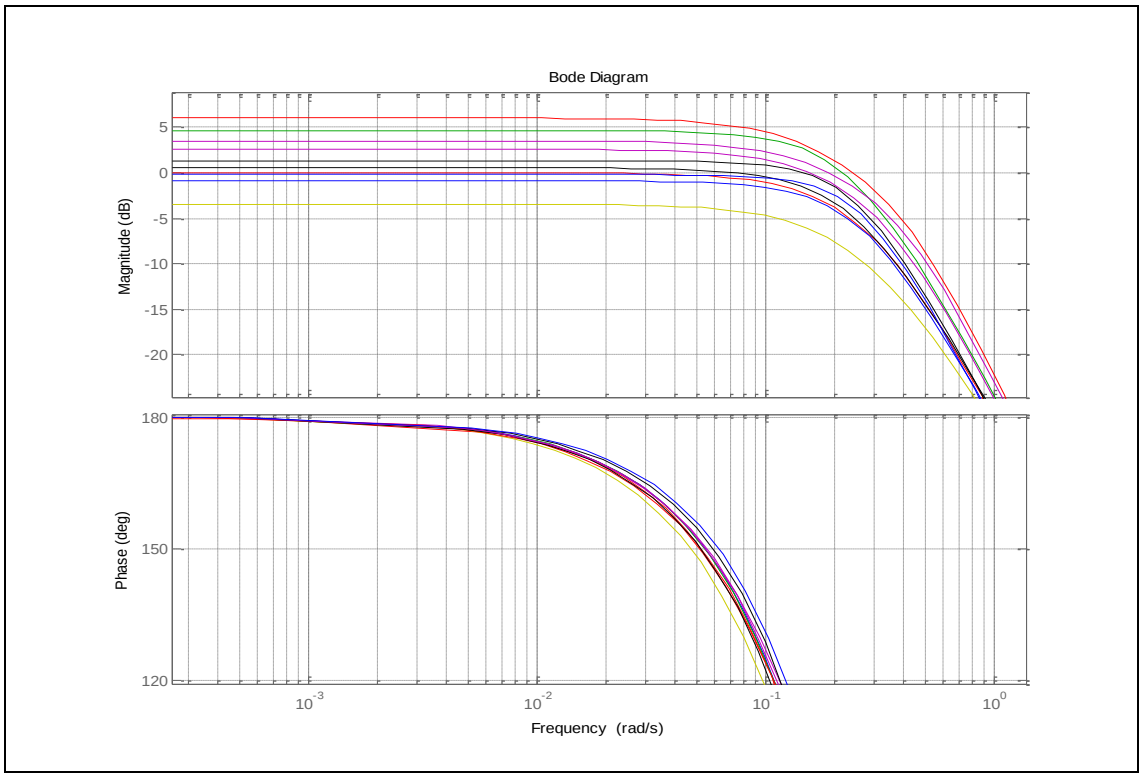


Figure 5-32: Bode plots of the uncontrolled plant for different parameter values

The step response obtained with the nominal plant are shown in figure 5.33.

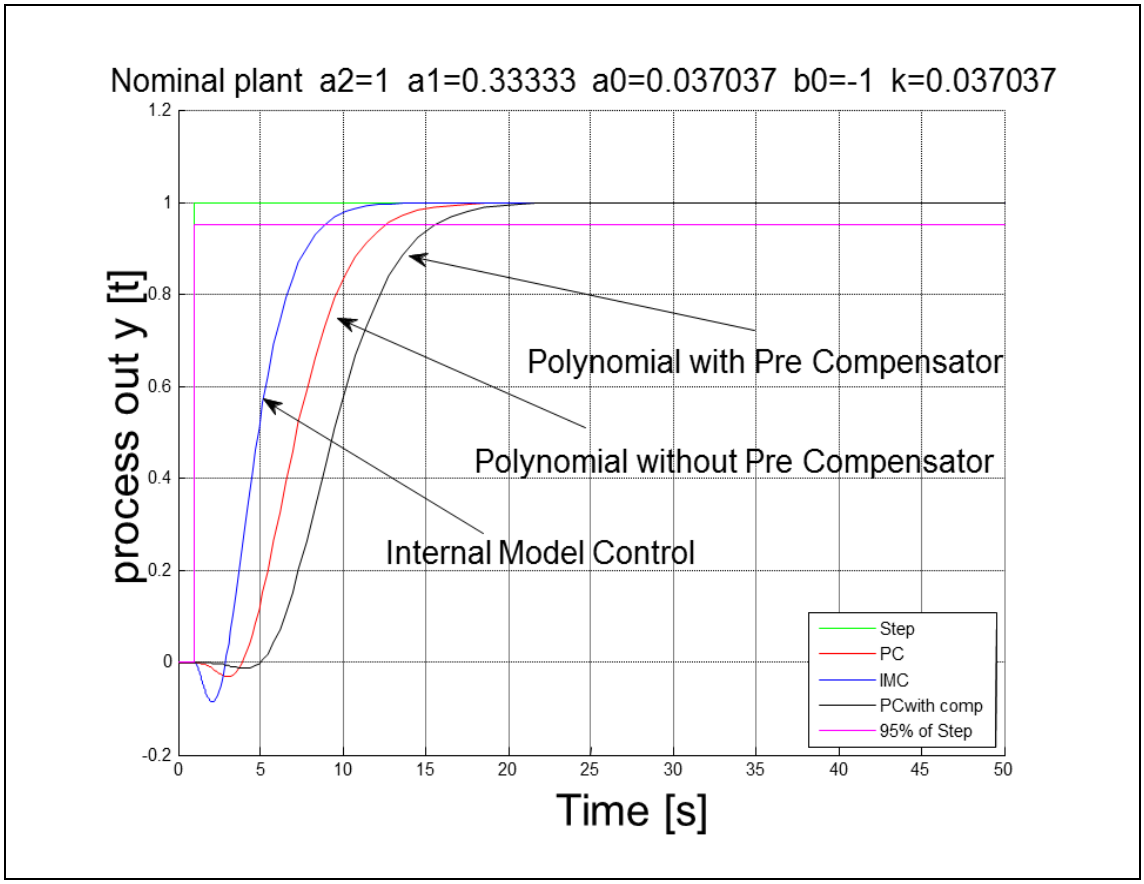


Figure 5-33: Step responses for the plant precisely matching the nominal plant model

Both control methods were subjected to the same plant mismatches and the results of the simulations are shown in Figures 5.34 and 5.35. It can be seen that the nominal performance of both controllers (IMC and RHF) is satisfactory and their closed loop response to a step input remains stable. The only significant difference that can be seen is at the start of the step response where the pre-compensator is minimising the effect of the zero's as shown in Figure 5.36 and they have different settling times. The response time of the RHF-controller is delayed because of the reference input signal being filtered. Clearly the RHF-controller. The IMC controller has stable control on the plant as shown in Figure 5.34 but the deviation against model uncertainty is considerable with large overshoots with extreme plant mismatches, in contrast to the RHF controller, which maintains the specified transient response specification very tightly with only a very small deviation from the nominal performance, as shown in Figure 5-35.

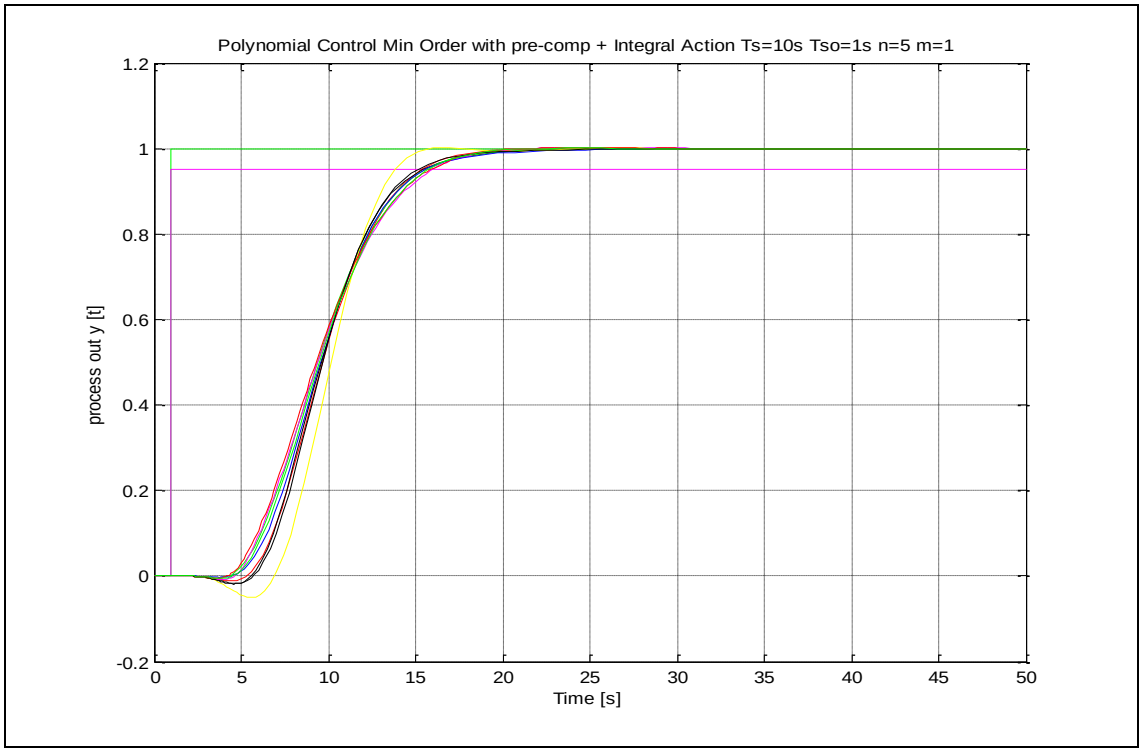


Figure 5-34: Step response of RHF with model uncertainty

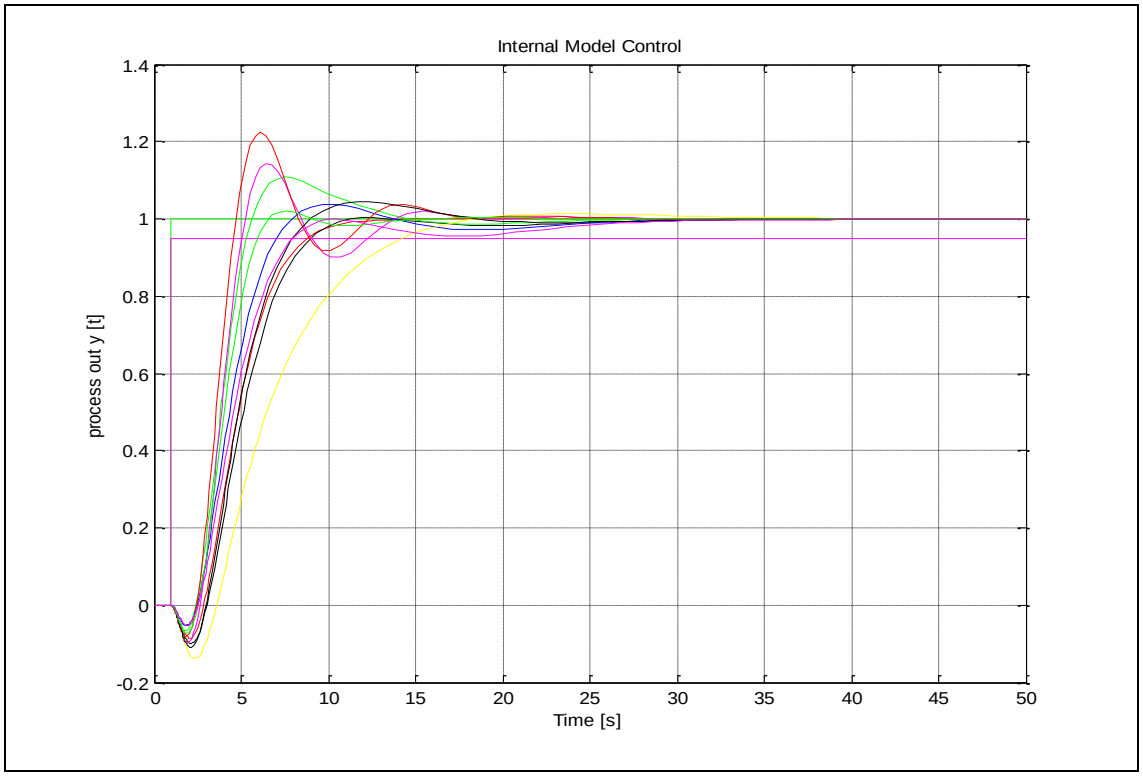


Figure 5-35: Step response of IMC with model uncertainty

Figures 5.36 and 5.37 show a simple comparison of the RHF controller structure with and without the pre-compensator, which shows in some cases the pre-compensator will be required to minimise the effects of the zeros in the closed loop system. The proposal here is to design the closed loop system first, then run some simulations and observe the effect of the zeros. In this way one can decide if a pre-compensator is required

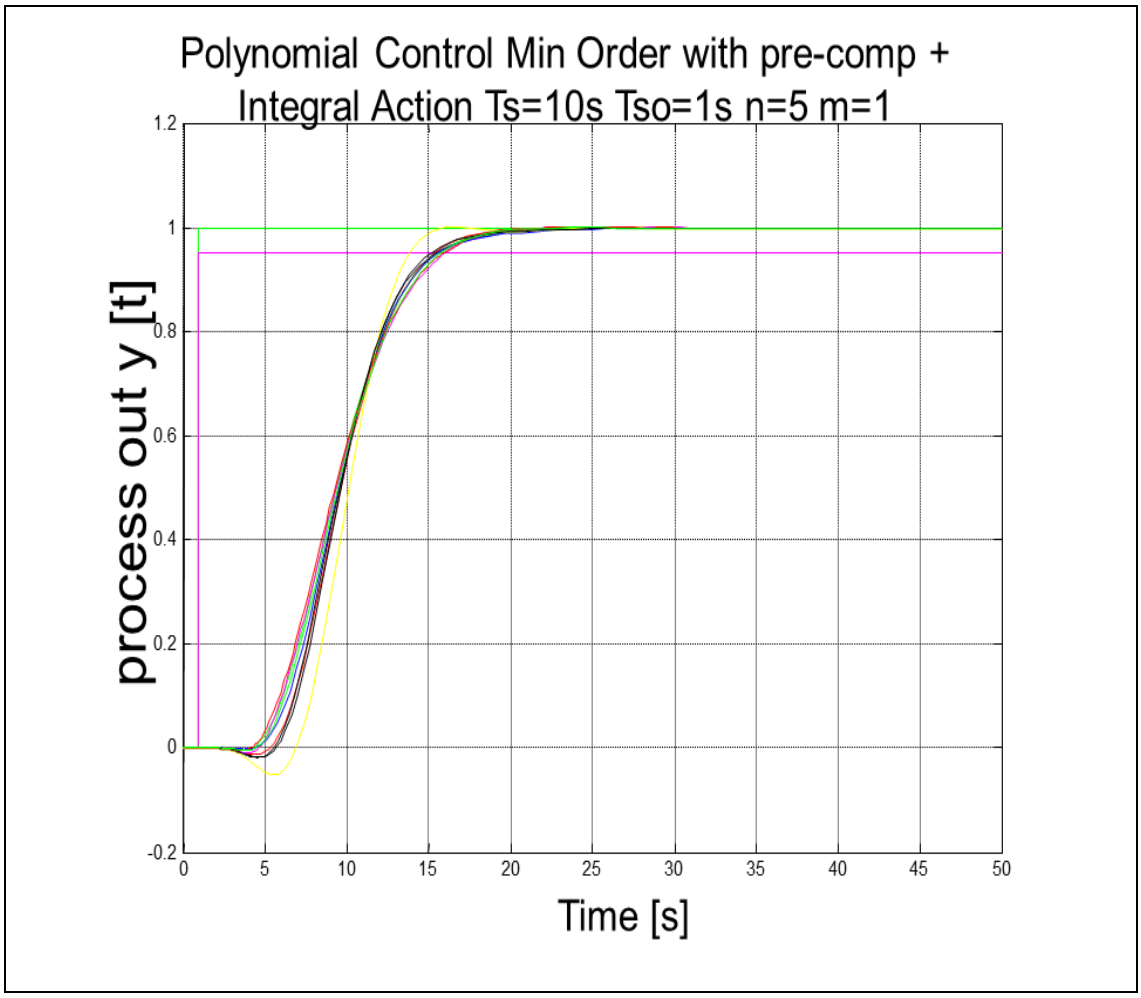


Figure 5-36 Step response of RHF with pre-compensator

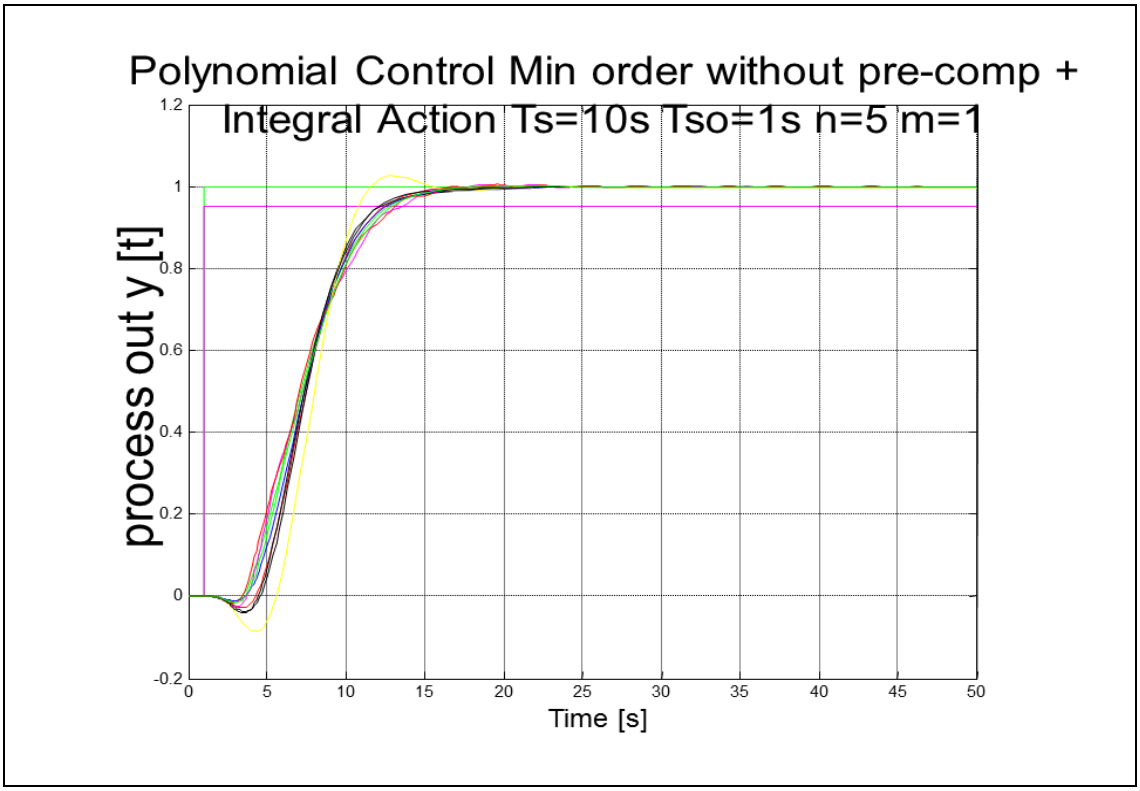


Figure 5-37 Step response of the RHF controller without pre-compensator

5.8.6 Comparison of RHF and RST Controllers

The results of the simulations are for two RHF-controllers, one with and one without a pre-compensator. The reference input consists of the sum of two sine waves, one at a frequency of 5 [rad/s] and the other at 7 [rad/s] respectively, and a linear ramp. In addition, a modified form of this reference input is applied with the sinusoid at 7 [rad/s] changed to one at 16 [rad/s]. The reason for this is that both the RFH and RST controllers were highly tuned to the first reference input and it is considered necessary to assess their performance with another reference input, and without retuning.

A pre-compensator is introduced with the RHF controller to compensate for dynamic lag of the closed loop system. In this case it is a second order system with transfer function,

$$\frac{R(s)}{Z(s)} = \frac{s^2 + 2\xi\omega s + \omega^2}{(T_{fc}s + 1)^2}, \tag{5.71}$$

where $\xi = 0.53$ and $\omega = 10$. The second order low pass filter element is included to enable straightforward implementation as a dynamical system, instead of software differentiation, such as in a dSPACE system using Simulink. Normally, the polynomial, $s^2 + 2\xi\omega s + \omega^2$, would be the same as the nominal characteristic polynomial of the closed loop system. The filter, however, would introduce a certain amount of dynamic lag, determined by the time constant, T_{fc}, but since this is outside the feedback loop it would not cause oscillations or destabilise the closed loop system.

For some applications, the ideal way of compensating for the dynamic lag of a closed loop system is pre-computation of the first and second derivatives of the reference input and feeding these forward together with the reference input with relative weightings of $2\xi\omega$ and ω^2, respectively. This would avoid the need for a low pass filter.

Figure 5.38 shows that the pre-compensator has minimised the error and compensates for the dynamic lag.

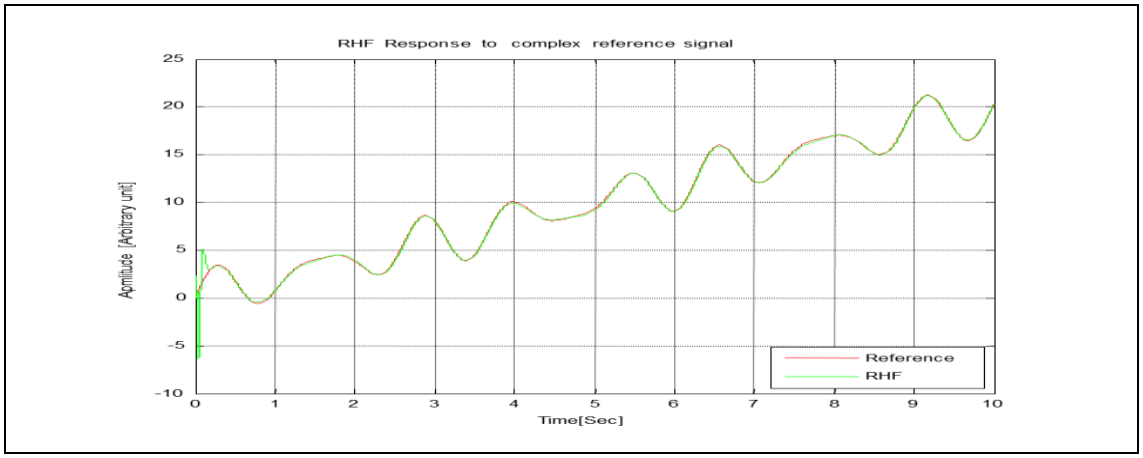


Figure 5-38: RFH with pre-compensator response to reference input (sine(7t)+sine(5t)+ramp)

Figure 5.39 shows the dynamic lag occurring without the pre-compensator.

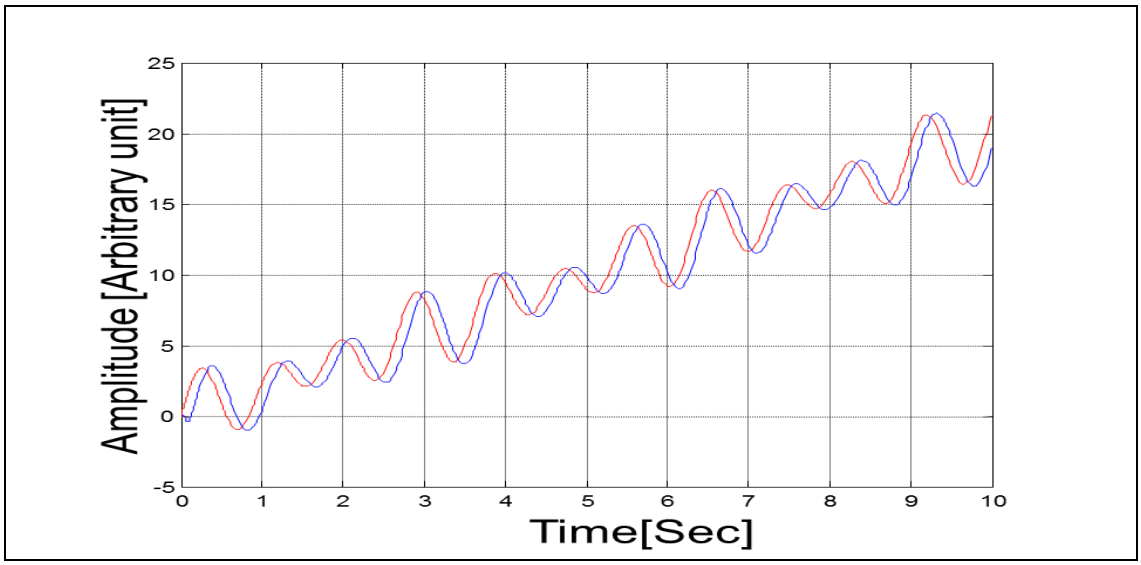


Figure 5-39: RFH without pre-compensator response to reference input (sine(7t)+sine(5t)+ramp)

Figure 5.40 shows the control error of the RHF controller with the pre-compensator.

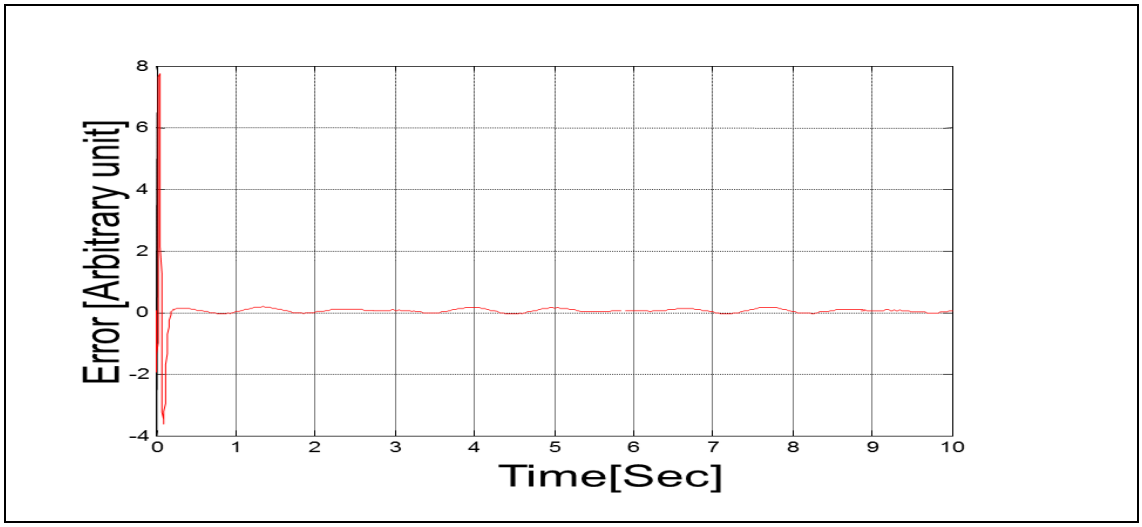


Figure 5-40: Error signal for RFH with pre-compensator

At the start of the simulation the controller has a large error, some this error is due to the solver algorithm and some of it is due to the initialisation of the controller parameters. Figures 5.38 and 5-41 show that the RHF and RST controllers are performing well with the complex reference input. Both controllers were highly tuned for the first reference

signal with 5 [rad/s] and 7 [rad/s] sinusoids. Figure 5.41 shows that the RST controller is following the reference signal well even when the 7 [rad/s] component is changed to 16 [rad/s] and the RHF follows the reference input slightly better than the RST controller.

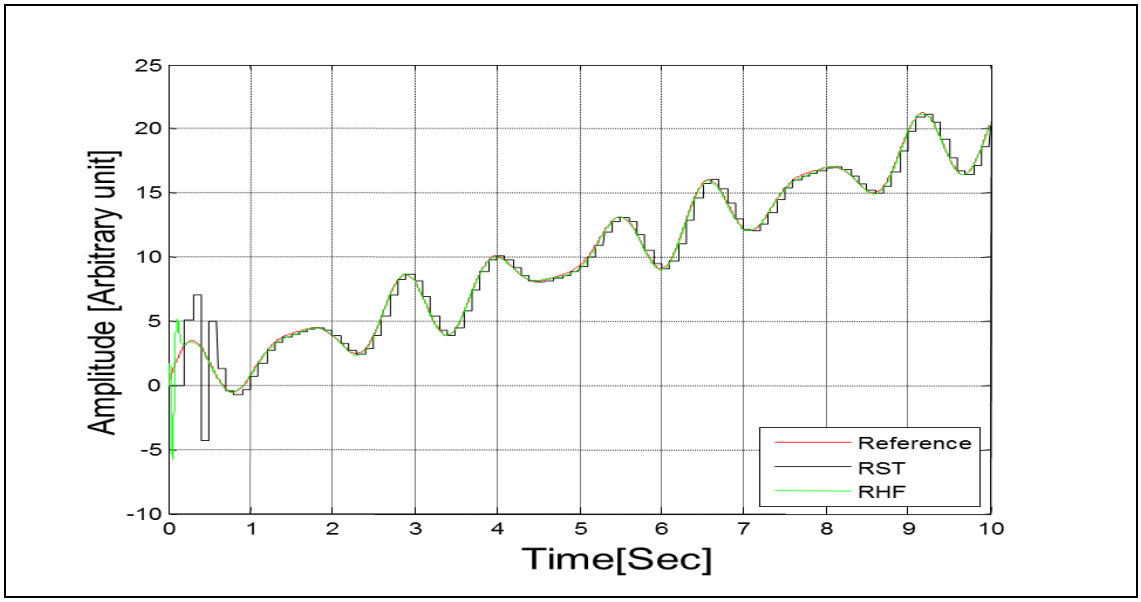


Figure 5-41: RFH with pre-compensator and RST response to reference input (sine(16t)+sine(5t)+ramp)

Figure 5.42 shows the control errors of the RST and RFH with and without the pre-compensator.

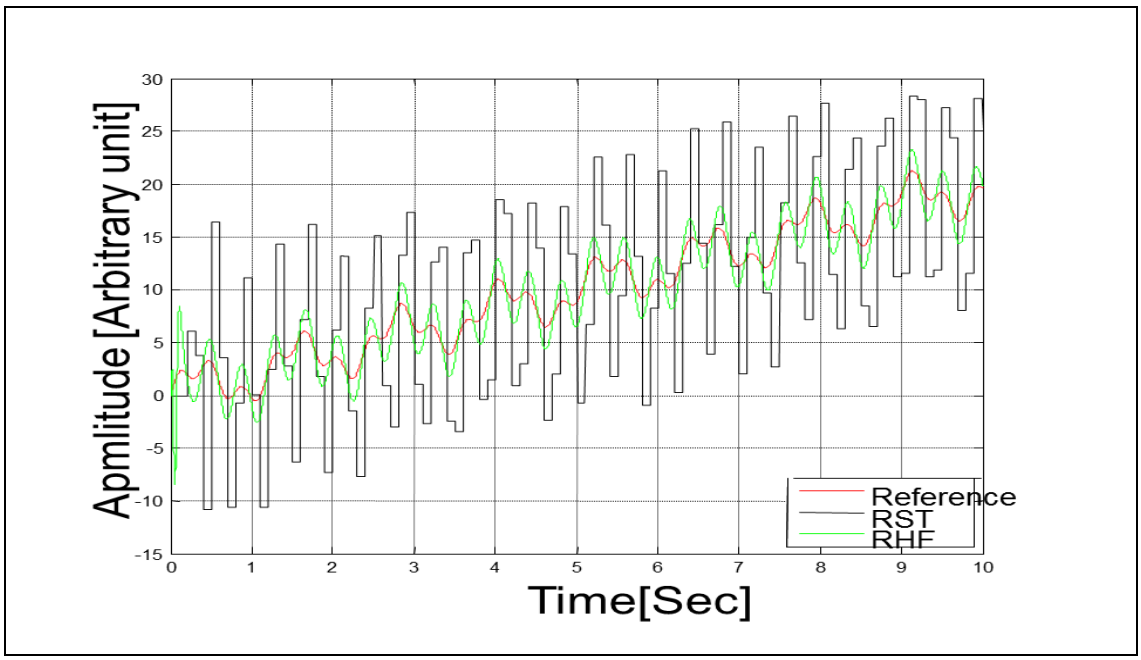


Figure 5-42: Error for RFH with pre-compensator and RST

The RHF controller effort, however, is much higher than that of the RST controller as shown in Figure 5.43, which is the price to pay for the smaller error.

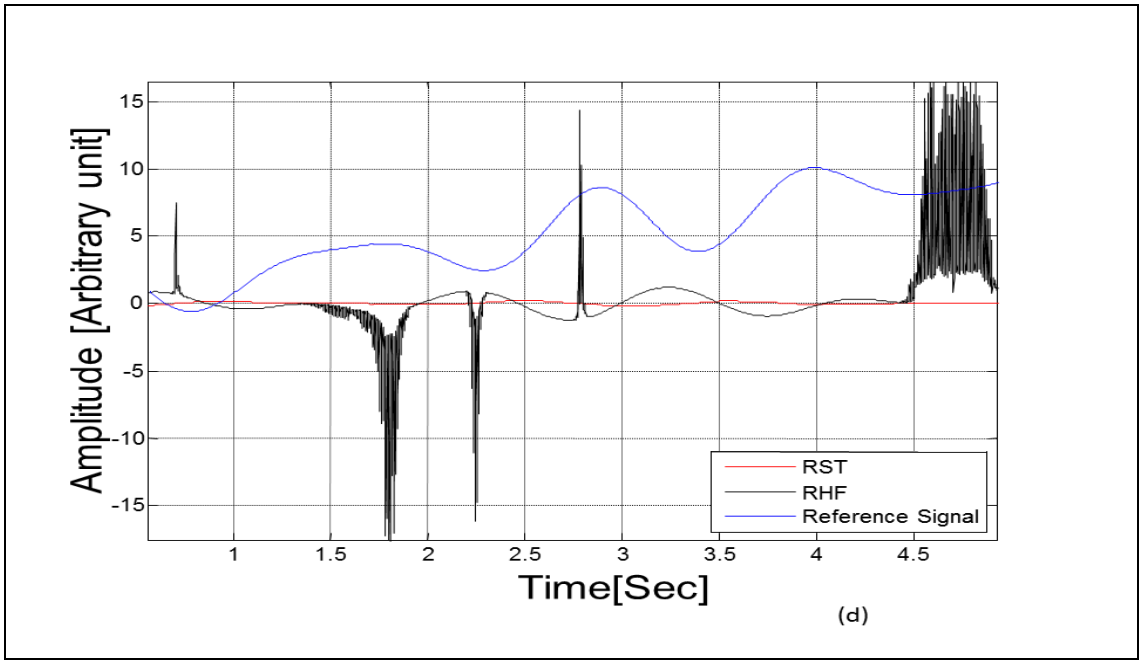


Figure 5-43: Plot of controller effort for RFH with pre-compensator and RST

5.8.7 Summary

Both controllers provide a robust solution for a given plant and the question is the degree of robustness which will depend on the quality of the linearised model and how close is the nominal model to the mean of the plant. Both controllers provide the designer with an adequate degree of adjustability and flexibility. Also the RHF design process is a simple as that of the IMC. This could also be said of the RST with similar approach to the RHF. H-infinity, however, and RST (at present) are using a sophisticated mathematical optimisation which will require a complex design procedure and high computation power. The degree of robustness which the controller can yield depends on the linearised model and the closed loop specification. As soon as an un-modelled element is introduced in the plant, the H-infinity controller must be redesigned based on new constraints, which is an expensive and time consuming process. In comparison, the OBRC, RHF, and IMC can be re tuned to certain degree. Also all of the design methods which are compared in this work provide robustness, but some are more time consuming and costlier than others.

6. Comparison of Control Techniques for Engine Application

6.1 Overview

In this chapter the IMC controller and OBRC controller performances are compared for the Diesel engine application. This comparison is basically by simulation but some experimental results are also presented for the OBRC controller.

6.2 The OBRC Controller

6.2.1 Overall Structure

Figure 6-1 shows the overall block diagram of the observer based control system for speed control of the Diesel engine including the external load torque.

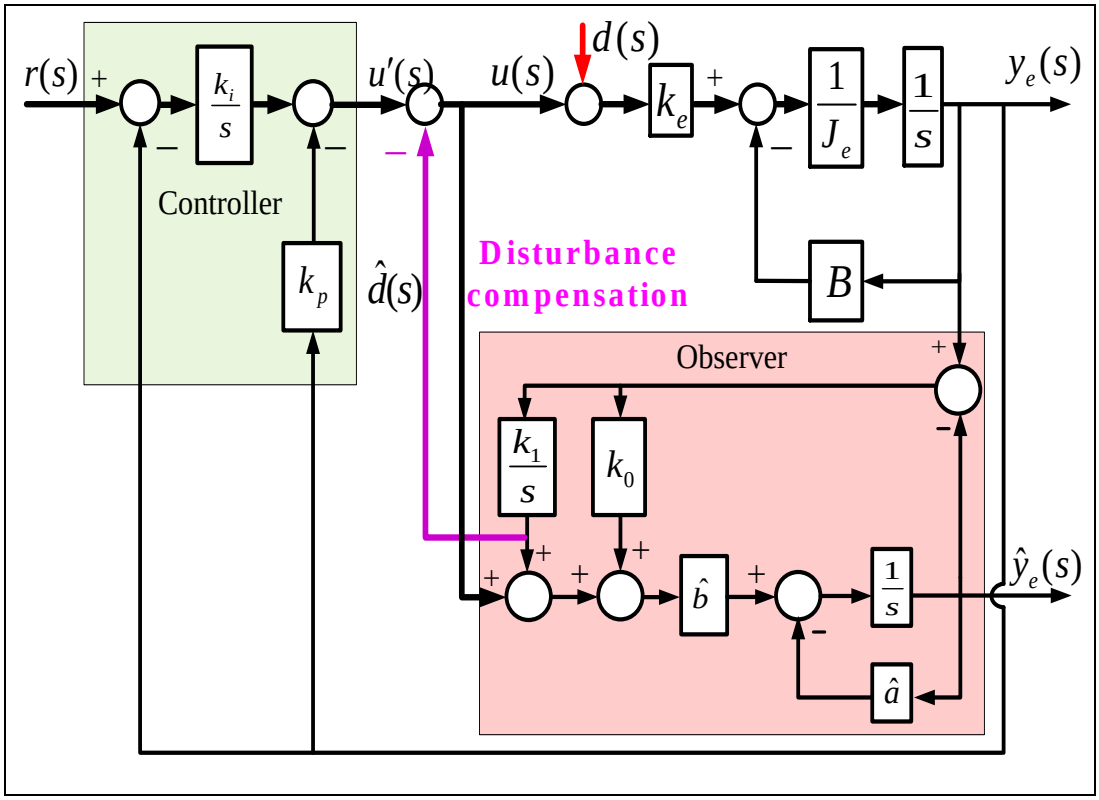


Figure 6-1: Block diagram of OBRC control system for Diesel engine

The model controller is chosen as an IPD controller since a) it is fairly well known in the industry and b) it is closely related to the universally used PI controller but, in contrast with this, does not introduce zeros in the closed loop transfer function and cannot, therefore, cause unavoidable overshooting in the step response. Its design is presented in the following subsection.

6.2.2 Design of the OBRC Model Controller

An IPD controller is selected for the OBRC model control. For ideal operation this will directly control the engine as shown in the simplified block diagram of Figure 6-1.

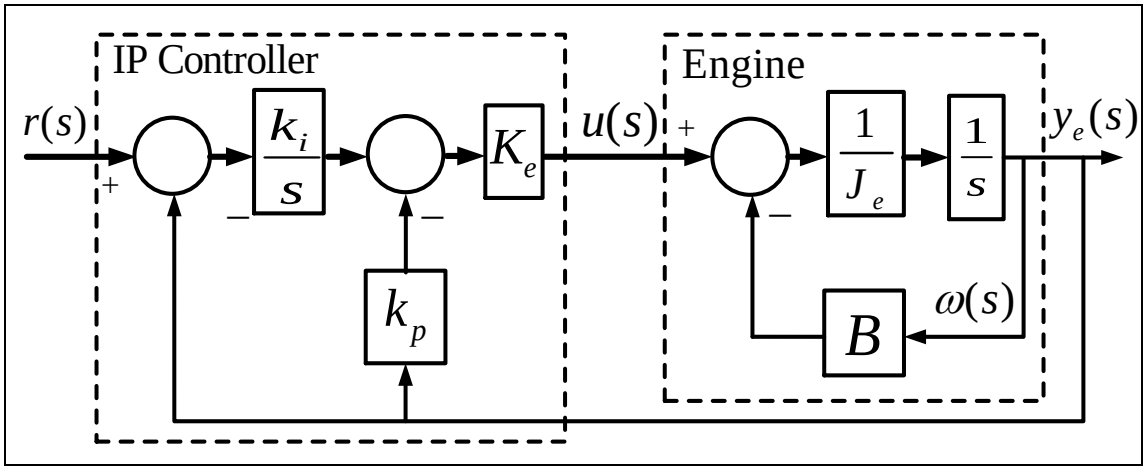


Figure 6-1: Closed loop System block diagram for engine speed control

This controller does not introduce a zero in the closed loop transfer function, due to the proportional term acting only on the controlled speed measurement, $y_e(s)$. In the PI controller, the proportional term acts on the error, $r(s) - y_e(s)$, giving rise to a zero in the closed loop transfer function and the possibility of an unwanted overshoot in the step response. The IP controller yields the same closed loop poles as the PI controller for given settings of the integral gain, k_i, and the proportional gain, k_p.

Calculation of the engine controller gains is based on pole assignment.

From Figure 6-2, the closed loop characteristic equation is given by equating the determinant of Mason's method to zero. Thus

$$1-\left[\frac{1}{J_e s}\left(B+K_e k_p+\frac{K_e k_i}{s}\right)\right]=0 \Rightarrow s^2+\frac{1}{J_e}\left(B+K_e k_p\right)s+\frac{K_e}{J_e}k_i=0 \qquad (6.1)$$

The closed loop poles will be made coincident at a location determined by the settling time formula (2.23). Equating the characteristic polynomial of equation (6.1) to the desired characteristic polynomial then yields

$$s^2+\frac{1}{J_e}\left(B+K_e k_p\right)s+\frac{K_e}{J_e}k_i=\left(s+\frac{1.5(1+n)}{T_s}\right)^n\Bigg|_{n=2}=s^2+\frac{9}{2T_s}s+\frac{81}{4T_s^2}$$

Hence

$$k_i = \frac{81 J_e}{4 T_s^2 K_e} \quad \text{and} \quad k_p = \frac{1}{K_e}\left(\frac{9 J_e}{2 T_s} - B\right) \tag{6.2}$$

T_s will be set to $1[\mathrm{s}]$. Substituting the value of K_e from equation (3.26) together with J_e and B from Table 3.1 then yields $k_i = 81$ and $k_p = 12.75$.

6.3 The IMC Controller

6.3.1 Overall Structure

Figure 6.2 shows the closed loop system for the engine application with the standard form of the IMC controller.

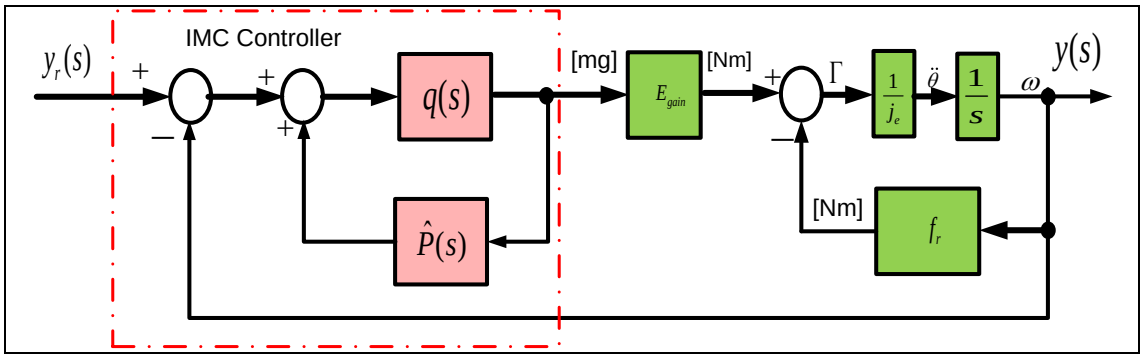


Figure 6-2 Closed loop block diagram incorporating IMC controller.

Figure 6-3 shows the equivalent block diagram in the unity feedback form.

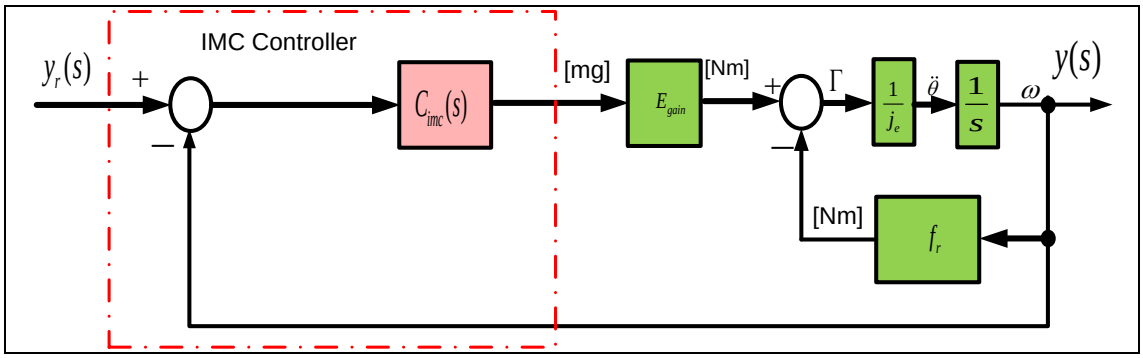


Figure 6-3 Closed loop block diagram of IMC structure with unity feedback

6.3.2 IMC Controller Design

The reader is referred to section 5.6 for the design equations. To determine the IMC controller gain, C_{imc}, in Figure 6-2, equation (5.55) yields

$$C_{imc} = \frac{G_p^+(s)C_f(s)}{1-G_p^-(s)C_f(s)} \tag{6.3}$$

where the $G_p^+(s)$ is the component of the plant transfer function that is invertible. The engine transfer function is completely invertible and yields

$$G_p^+(s) = \frac{(s+\tau_e)}{k_m} \ . \tag{6.4}$$

So, to fit in with this general design methodology, the non-invertible factor is given by

$$G_p^-(s) = 1 \tag{6.5}$$

Also the $C_f(s)$ is the filter component of the controller which has the same order the as the plant $(n=1)$. Therefore

$$C_f(s) = (\tau_f s + 1) \tag{6.6}$$

Substituting for $G_f(s), G_p^+(s)$ and $G_p^-(s)$ into the equation (6.3) then yields

$$C_{imc}(s) = \frac{\dfrac{(s+\tau_e)}{k_m}\dfrac{1}{(\tau_f s+1)}}{1-(1\cdot\dfrac{1}{(\tau_f s+1)})} \Rightarrow$$

$$C_{imc}(s) = \frac{1}{k_m}\frac{(s+\tau_e)}{(\tau_f s)} \tag{6.7}$$

where $k_m = 2\tau_f$ and $\tau_e = 30$. Hence (6.7) becomes

$$C_{imc}(s) = \frac{1}{2\tau_f}\frac{(s+30)}{s} \tag{6.8}$$

$$C_{imc}(s) = (s+30)\frac{k_{imc}}{s} \tag{6.9}$$

where $k_{imc} = \dfrac{1}{2\tau_f}$.

Now, the only adjustable parameter that is left is the filter time constant τ_f . The final value of the filter time constant is determined experimentally, since the initial value may not provide an acceptable response. Figure 6.4 shows the implementation of the IMC controller. The controller is divided into two parts.

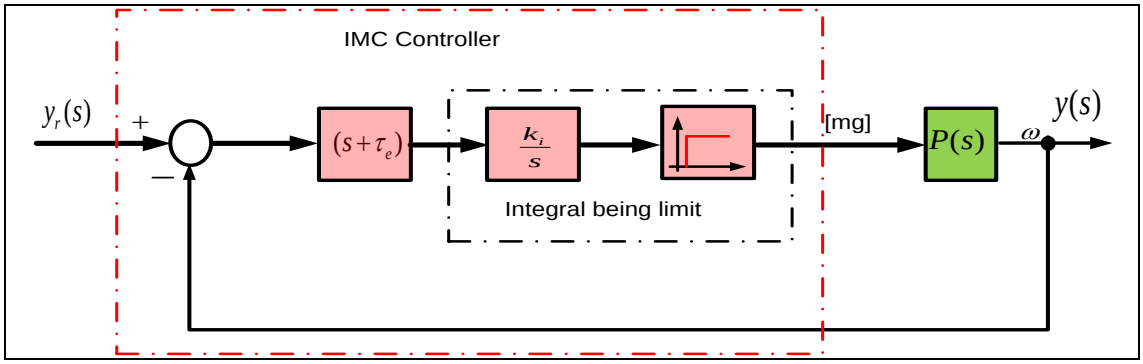


Figure 6-4: Basic IMC engine control loop

The first part is the integral term which is bounded to the plant limits (0 - 350 [mg/ stroke]), as shown. The second part is implementation of the derivative term, which has inbuilt low pass filtering to avoid amplification of high frequency components of measurement noise. The implementation of this is shown in Figure 6.5.

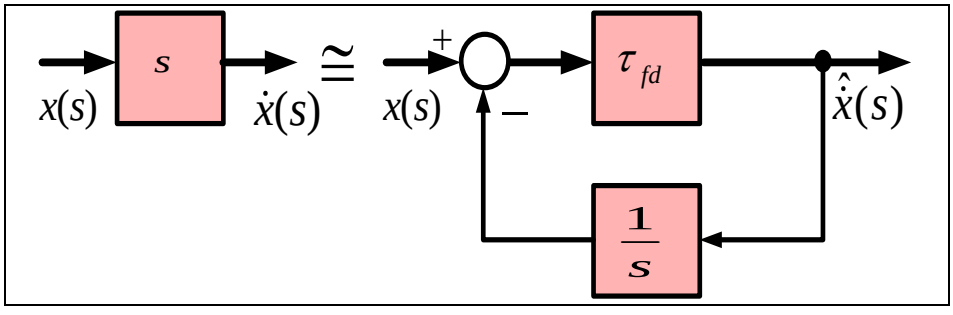


Figure 6-5: Derivative block with inbuilt low pass filtering

The transfer function of the of the filtered derivative block of Figure 6-12 is

$$\frac{\hat{\dot{x}}(s)}{x(s)} = \frac{\tau_{fd} s}{s + \tau_{fd}}.$$

It should be noted that, in contrast to most of the other control techniques considered, the IMC design methodology does not allow the control engineer to work to or specify a transient profile for the step response, but instead aims at the ideal of zero dynamic lag. Figure 6.6 shows a pre-compensator, which is introduced to obtain a transient response comparable with that of the OBRC and to avoid immediate control saturation when step changes in the reference input occur.

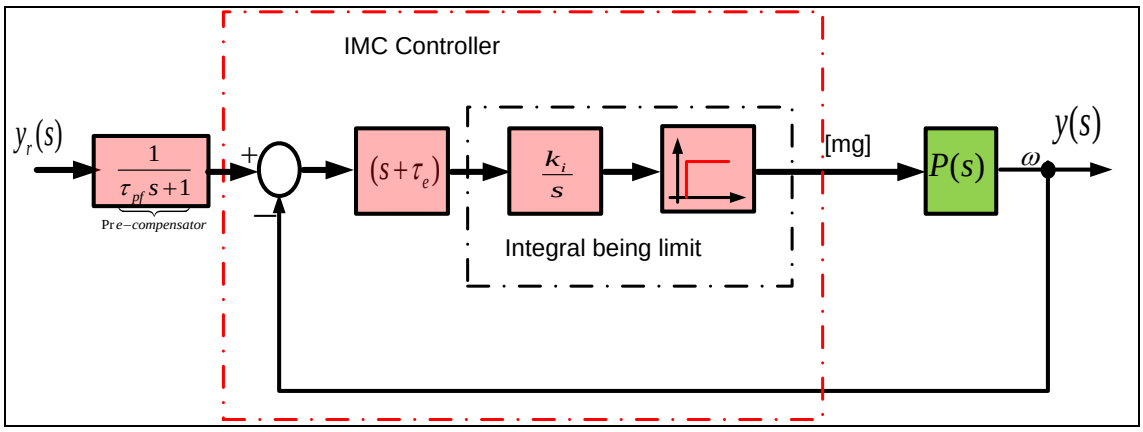


Figure 6-6: IMC engine control loop with pre-compensator

It should be noted that the first order pre-compensator yields an overall transient response that is first order in character. If necessary, a more sophisticated pre-compensator could be designed to yield other forms of overall transient behaviour.

6.3.3 Simulations

The simulations were carried out in the Matlab/Simulink environment based on the experiment results collected from test runs carried out on the engine. The comparison would be against the data collected for the OBRC controller on the engine. The OBRC simulation was validated against the engine result that was used for the IMC controller and the OBRC controller.

Figures 6-7 to 6-9 show results with the IMC controller with and without the pre-compensator. It can be seen that a simply first order filter can reduce the controller activity and remove the harsh response of the controller and instead provide a pre-defined step response profile.

Also it is clear that the gradual change of the control input fuel afforded by the version employing the pre-compensator is very desirable and any large change in the fuel quantity will create larger acceleration, which is not desirable.

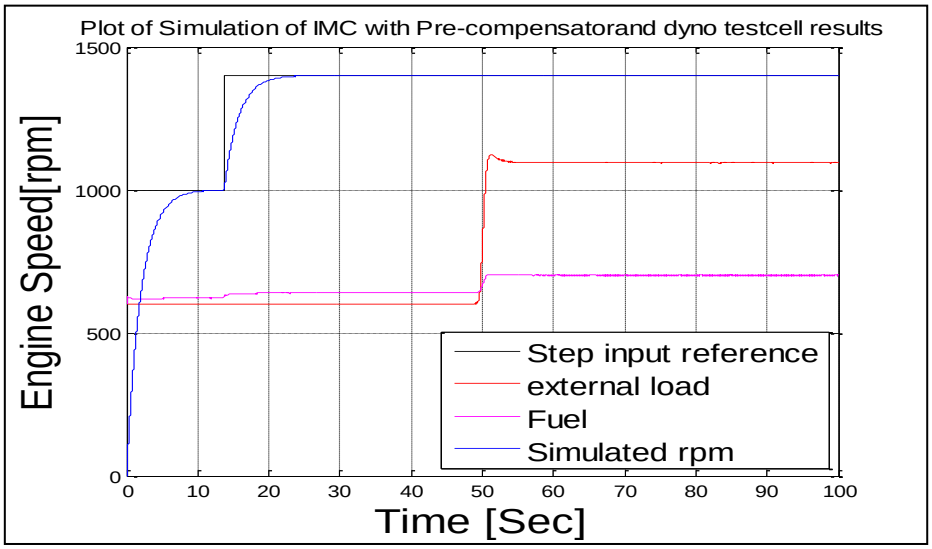


Figure 6-7: Step response of the IMC controller with pre-compensator

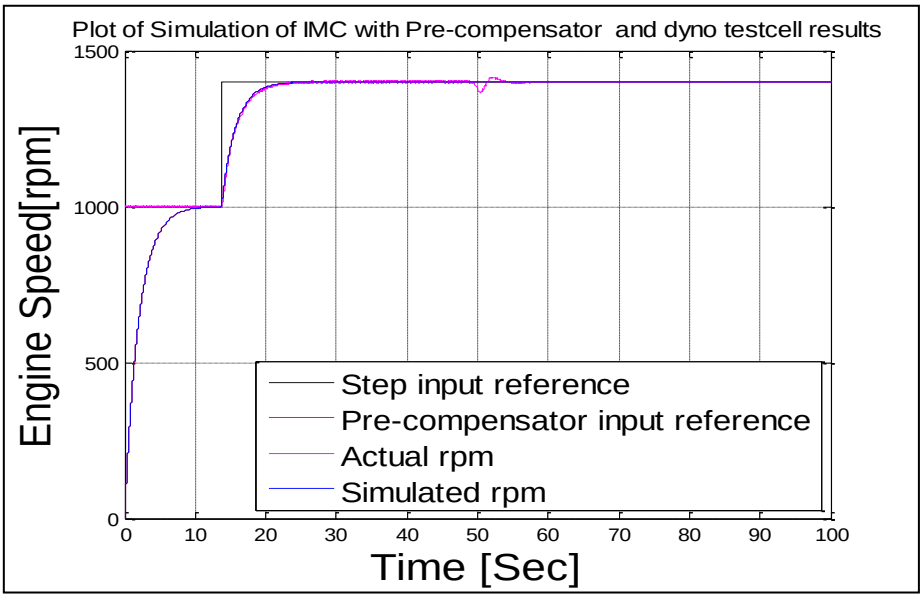


Figure 6-8: Further step response of the IMC Controller with pre-compensator

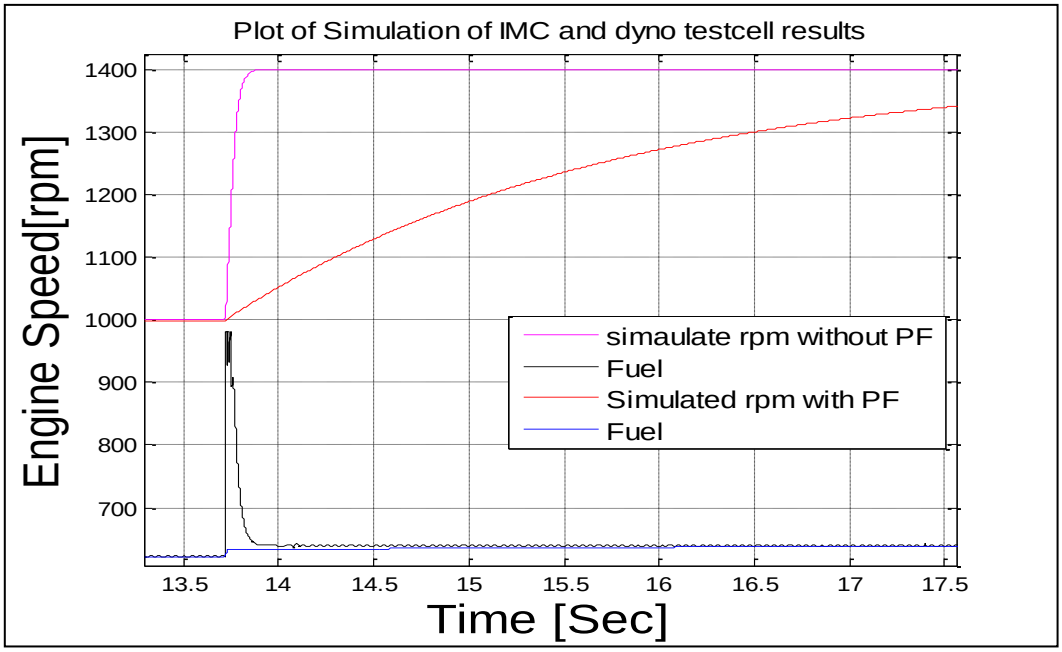


Figure 6-9: Step response of the IMC with and without pre-compensator

6.4 Experimental Setup

The OBRC engine control system was implemented in Matlab/Simlink using the dSpace system desk. The observer and controller were developed with the dSpace Micro-Autobox. The Micro-AutoBox was connected to the EMS via a CAN link. The torque signal from dynamometer system was connected to the Micro-AutoBox via a 16bit ADC channel. The engine test cell had a dynamometer of the eddy current type with no motoring capability). There are several options for the dynamometer control mode to be set, including speed control and torque control. Throughout the experiment, the dynamometer was setup in the torque control mode control. In this mode, the dynamometer control system cannot change the load unless requested by the operator or changing to the speed control mode. Figure 6.17 shows an overview of the experimental setup. There were two PC's, one connected to the Micro-AutoBox and the other connected to the EMS for monitoring of the EMS measurement signal and modifying the calibration parameters.

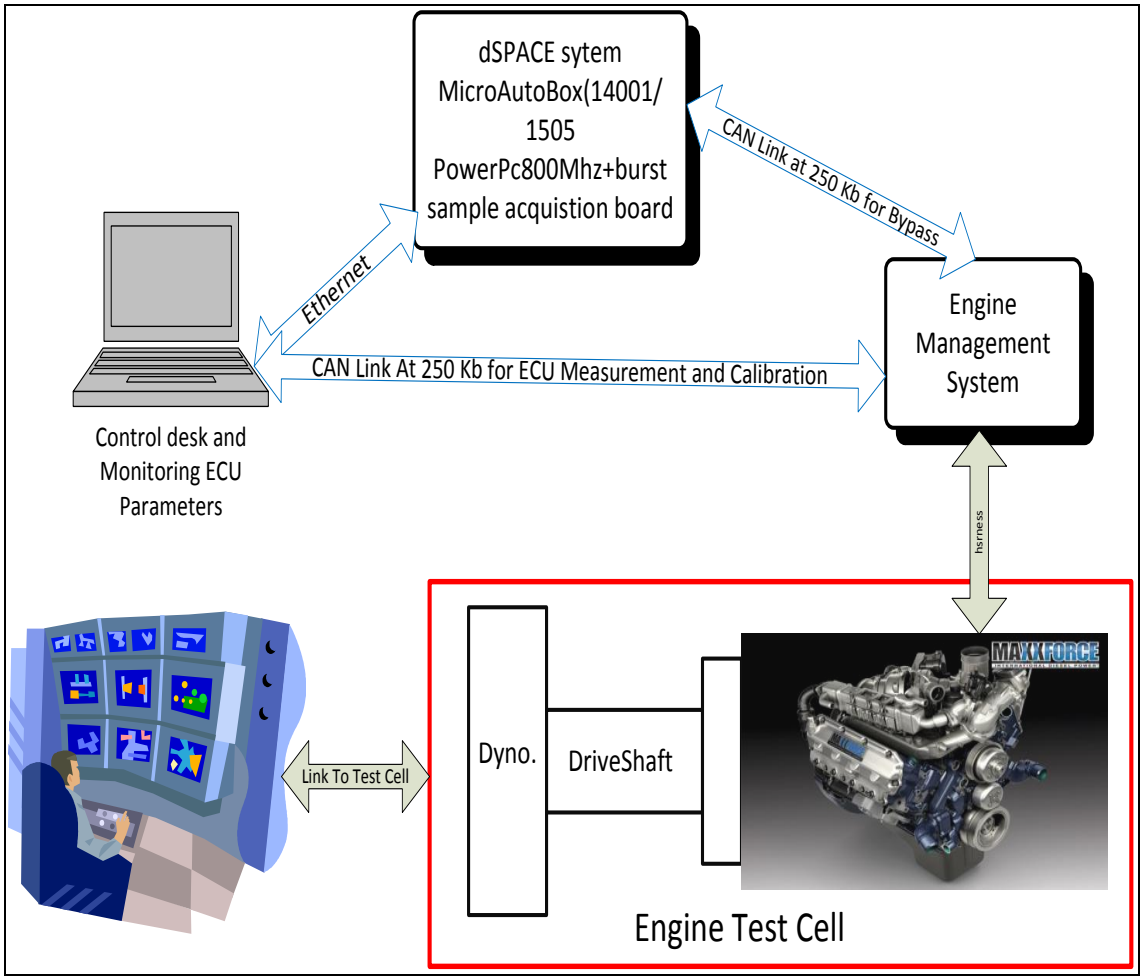


Figure 6-10 Experiment Setup

6.5 Comparison of Simulation Results and Test Results

The data collected from the test cell was replayed in the Matlab environment. The simulation was run under the same conditions as the test cell with the demanded engine speed as a common input. After each simulation run, the results were compared with the corresponding data collected from the test cell. In Figure 6-11, the physical system and the simulation commence with the control system operating in the steady state with a reference speed input of $500[\text{rpm}]$ and zero load torque. A step increase of the reference speed input to $1000[\text{rpm}]$ is applied at $t=0[\text{s}]$. This is followed by a further step increase in the reference speed to $1400[\text{rpm}]$ at $t=14[\text{s}]$. This reference input is maintained thereafter but a step load torque of $500[\text{Nm}]$ is applied at $t=50[\text{s}]$. It can

be seen that the simulated engine speed is a good match with the measured actual engine speed. It should be noted that the simulation was fed with the reference input and the measured torque from the engine experiment to ensure a fair comparison.

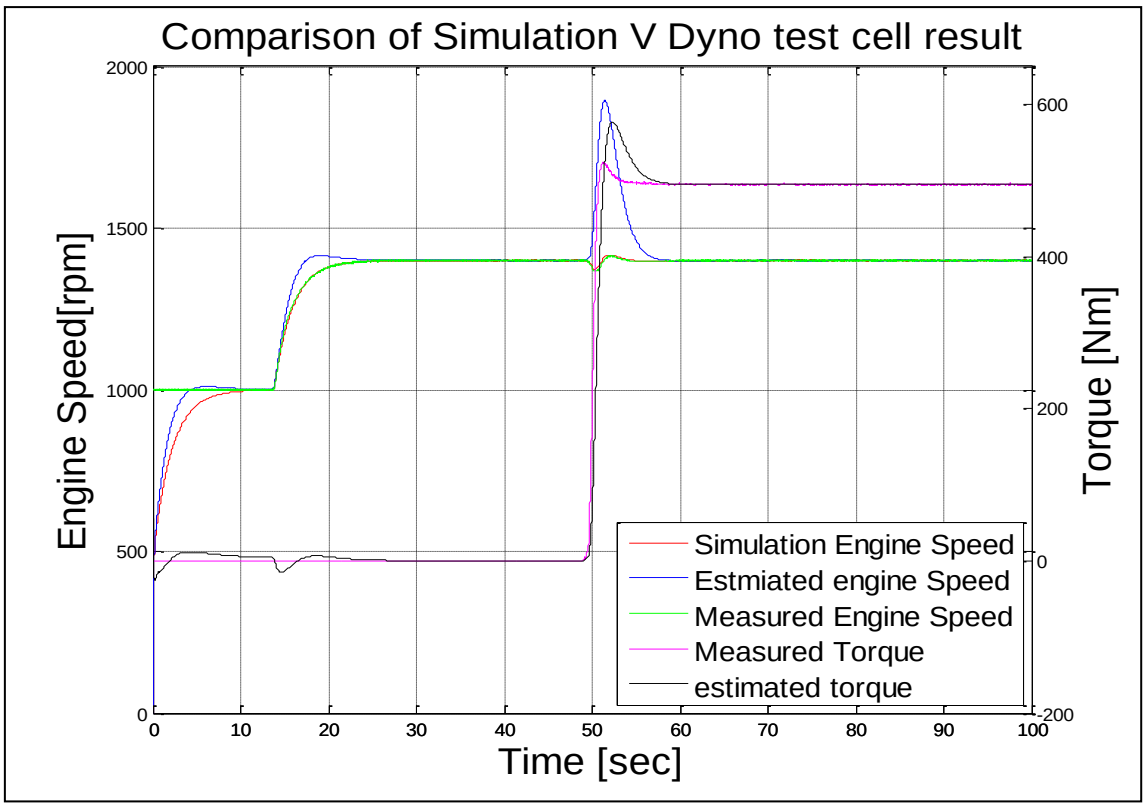


Figure 6-11 Comparison of simulated and test cell data

As expected, the control system responds with zero steady state error in the speed due to the integral term in the controller. The transients occurring in the estimated load torque following the speed reference steps occur on the time scale of the main control loop rather than that of the observer (recalling that $T_s = 1[s]$ and $T_{so} = 0.01[s]$). This is due to the time varying component of the estimated load torque estimate that compensates for any plant parameter mismatches, indicating that there are significant modelling errors.

Figure 6.12 shows the simulated torque from the model and actual torque measured on the dynamometer in the engine test cell.

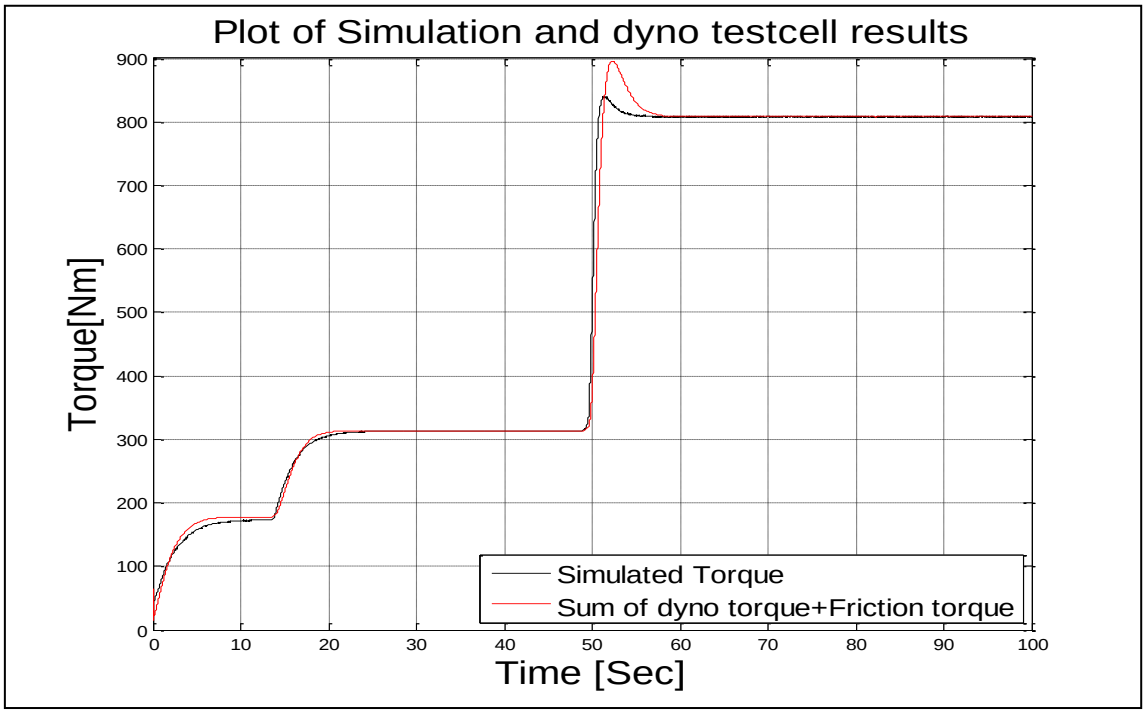


Figure 6-12 Comparison simulated torque and actual torque

The reference input to the simulation was the same as the reference input to the test cell. It can be seen from the Figure 6.20 that the simulation result follows the actual engine speed. The data was recorded during the experiment and subsequently used. The torque trace from the experiment was played back during the simulation. This was done by using a lookup table.

Figure 6.20 shows the simulated engine speed and measured engine speed. The experiment started at 1000 rpm and a step in speed was demanded (1000 rpm to1400 rpm). The overshoot is due to the observer load estimation and if the external load $\hat{d}(s)$ as shown in Figure 6.14 is removed from the control input then the control input becomes $U(s) = u'(s)$. The OBRC will undershoot due to the applied torque but will not overshoot as shown in figure 6.13.

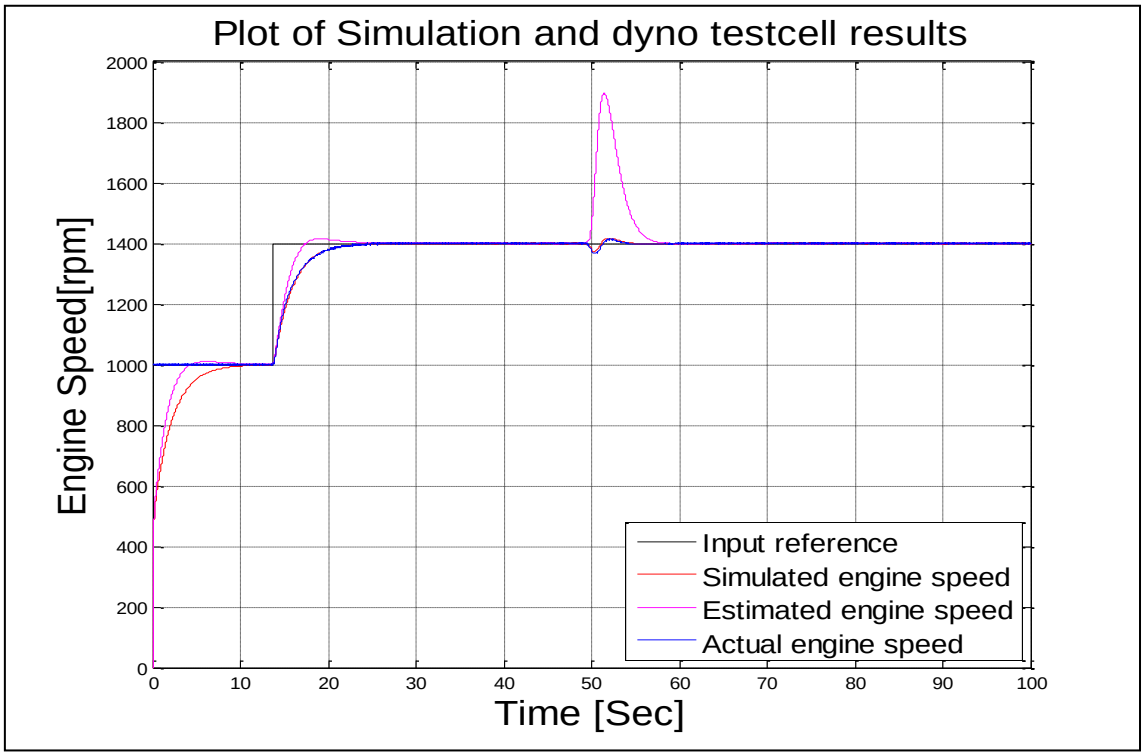


Figure 6-13 Comparison of actual engine speed with simulated engine speed

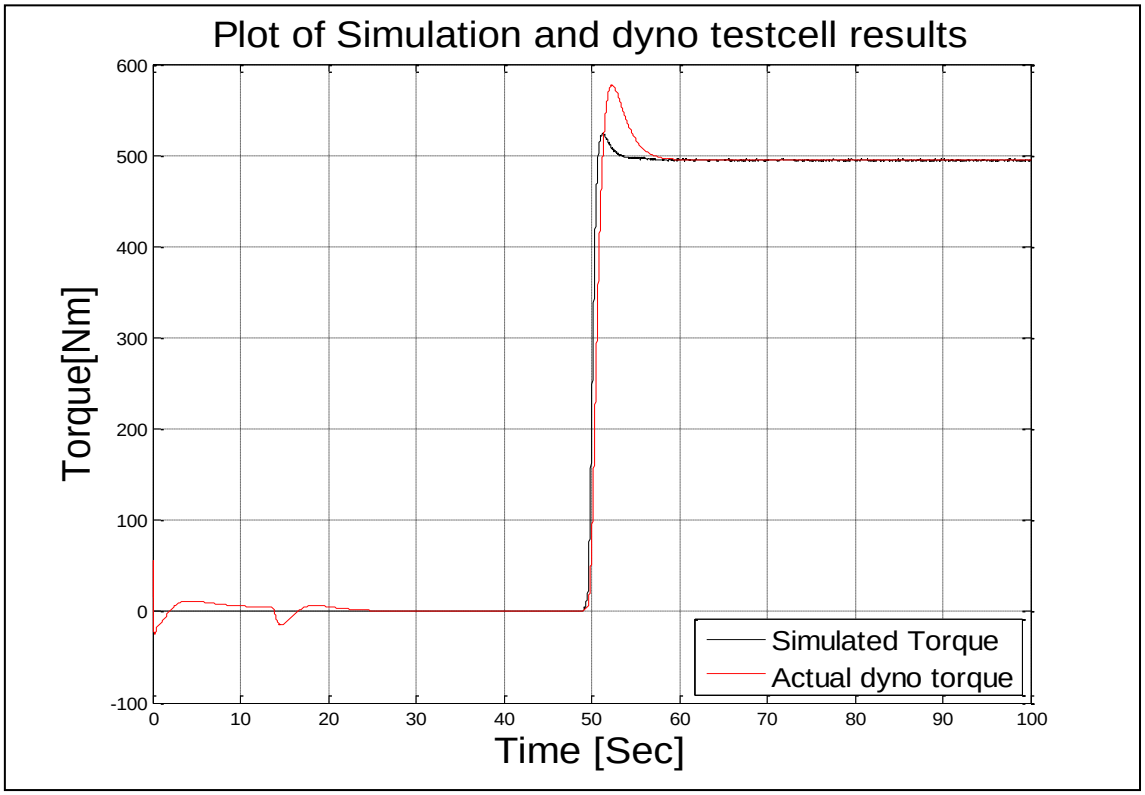


Figure 6-14 Comparison of the torque and estimated

Finally, to demonstrate that the feedback of the load torque estimate in OBRC is the key to its robustness, Figure 6-15 shows two responses of the control system with a constant reference input speed and a step external load torque applied at $t = 49.5[\text{s}]$. It is evident that this feedback reduces the magnitude of the transient speed error by an order of magnitude.

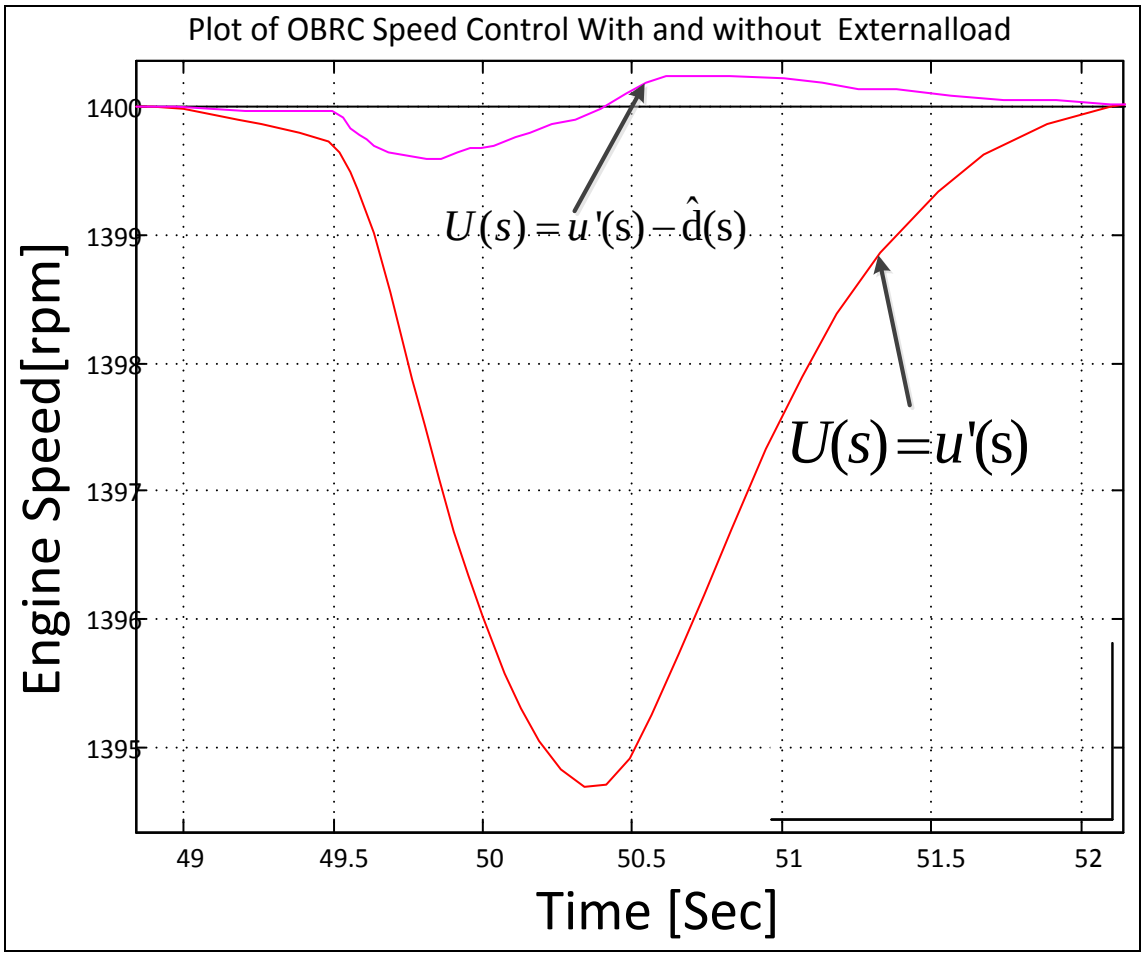


Figure 6-15: Plot of OBRC with and without external load signal

7. Conclusions and Recommendations for Further Research

7.1 Conclusions

7.1.1 OBRC

The OBRC methodology works very well for the class of plants excluding zeros in the right half plane and pure time delays. The inverse dynamic modeling technique allows a relatively complex plant to be modeled as two subsystems, a simple one in the forward path and one that is more difficult to model accurately, which is situated in the feedback path to the plant input. The effects of this feedback path subsystem are then represented by an equivalent disturbance input that can be estimated in an observer, using just an integrator, together with the state of the simple forward path subsystem. This work revealed a useful potential application in which the observer is used as a disturbance estimator to gain useful information, such as the load mass carried by a vehicle. Without this aid, the load torque would have to be measured by highly expensive torque transducers or complex modeling of the vehicle system. The main issue with OBRC is that it effectively cancels any zeros with closed loop poles automatically and therefore creates unstable closed loop modes if any of the zeros lie in the right half of the s-plane. Analysis of the OBRC during its development for plants of order greater than two, especially those with zeros, entailed very labor intensive derivations of the complete system characteristic polynomial for gain determination. As a result of this, a computer aided method was developed to determine the gains of any linear system for pole placement in which the relationship between the gains and the coefficients of the characteristic polynomial is linear, as shown in the Appendix. The Dodds settling time formula was used for the observer design and the model state controller design for the OBRC controllers. In the process of doing so, it was realised that this is a useful tool for any linear system that can be designed by pole placement.

7.1.2 Comparison of OBRC with Other Control Techniques

H-infinity controller

The basic OBRC controller was compared with an H-infinity controller, and the results were much the same for linear operation without control saturation. The difference was that the OBRC controller would produce more control activities at high frequencies. This is considered to be undesirable, from a practical aspect of the plant (actuator chattering and wear over the life the actuator). In view of this, a modified a low pass filter was introduced in the OBRC structure. The low pass filter limits the controller effort at high frequency and brings the frequency response of the $|k(s)\,S|$ for the OBRC into line with the H-infinity controller, and the results were still comparable. The OBRC controller, however, was able to cope with control saturation while the H-infinity controller was unable to do so.

Polynomial Controller, RST and IMC Controller

The RHF-controller has identical structure to the RST-controller. However, the two differ in the documented design approaches as a result of their independent evolvement, and also the fact that the RHF and RST controllers are formulated, respectively, in the continuous Laplace and discrete z domains. The RHF-Controller comes with a simple design methodology for full pole placement. An academic example was used to compare the RHF and IMC controllers. A parametric study was carried out and the result of the simulation and frequency analysis showed that the RHF-controller is more robust to model uncertainty than the IMC controller, with the model parameters mismatched relative to the plant parameters by up to $\pm 30\%$ for 10 simulation runs. The plant of the second academic example incorporated a pure time delay. The RHF-Controller provided good simulation results for a perfectly matched plant model but the degree robustness was reduced to a marginal point. In this case a pre-compensator had to develop to minimise the phase lag of the plant. Also it was shown when the reference input was changed significantly from 7 rad/s to 14 rad/s then both controllers start to lose tracking of the reference signal. Also, the RST controller was less robust than the RHF-controller.

7.1.3 Engine Test

The experimental test results showed that the robust controllers allow the possibility of eliminating the tuning process, but dependent on the accuracy of the available plant model. In the case of the diesel engine speed control, the model is very simple and no major issue is experienced in getting the design right first time. During these tests, nonlinearities were inserted into the plant model of the OBRC observer and the controller, but this did not require redesign of the controller or the observer correction loop gains for the OBRC. The torque estimation from the observer was compared with measurements from a torque transducer revealing an error of less than 2% over a large range of torques (0 to 1400 Nm). The OBRC was the only controller employed in these tests because the other control techniques did not provide any means of estimating the external load torque.

7.2 Overall Assessment

The simulation results, engine experimental work and theoretical investigations combine to confirm that generally more than one approach to solve a control problem may be successful. Some methods, of course, can handle plants better than others. The RHF, OBRC and H-infinity controllers proved to be the most robust, but with the reservation that the H-infinity controller could not handle control saturation. The IMC stands alone in that it can handle pure time delays in the plant. In terms of engineering cost and effort with modern software packages, as influenced by simplicity of the design process and accessibility to many industrial users, one would rank the RHF, IMC and OBRC-controllers more or less equally. In contrast, the RST and H-infinity controller design processes are complex and require sophisticated software tools and staff with a strong mathematical background to undertake in-depth development work. Although it is not essential to apply these associated computer aided design tools, however, dealing with issues arising in a particular control system development programme would be difficult without them.

7.3 Recommendations for Further Research

The research programme has revealed the following further investigations

1. To investigate the possibility of translating the z-domain based RST design methodologies to the Laplace domain for direct use with the RHF controller, and the identification of any advantages over the simple pole placement.

2. The development of a discrete version of the RHF controller that should be applicable to plants containing pure time delays

3. The design of reference input generators that enable derivative feed-forward for dynamic lag compensation to be carried out without the need for higher order software differentiation

4. Modification of the OBRC structure to enable plants with zeros in the right half of the s-plane to be controlled without the creation of unstable closed loop modes due to the zero cancellation inherent in the present structure. This alone would yield overshoots and/or undershoots of the step response in certain cases, requiring the design of pre-compensators outside the feedback loop using special pole placement techniques such as zero mirroring [19].

APPENDIX

A.1 Sensitivity Function

This section will define the sensitivity function for a given closed loop transfer function. Let us consider a non-dynamic system with a unity feedback structure having a forward path gain of G_1 and feedback gain, G_2. The loop gain is G_1G_2 and, if we assume negative feedback, the closed loop gain is given by

$$T = \frac{G_1}{1+G_1G_2} \tag{A.1}$$

In the real plant the parameter G_1 may vary throughout the plant life due to wear and operating conditions. The change in G_1 is denoted by ΔG_1 and the relative change is given by $\frac{\Delta G1}{G1}$. The corresponding change in the T is ΔT and the relative change is given by $\frac{\Delta T}{T}$.

The sensitivity function $S_{G_1}^{T}$ is defined as

$$S_{G_1}^{T} \triangleq \lim_{\Delta G_1 \to 0} \frac{\frac{\Delta T}{T}}{\frac{\Delta G1}{G_1}} = \lim_{\Delta G_1 \to 0} \frac{\frac{\Delta T}{\Delta G1}}{\frac{T}{G_1}} = \frac{\partial T}{\partial G_1}\frac{G_1}{T} \tag{A.2}$$

To obtain the expression for the sensitivity function we need to determine $\frac{\partial T}{\partial G_1}$ from the closed loop transfer function A.1. To differentiate the closed loop transfer function we need to use the quotient rule which is given by

$$\frac{\partial T}{\partial G_1} = \frac{\partial}{\partial G_1}\frac{u}{v} = \frac{v\frac{\partial u}{\partial G_1} - u\frac{\partial v}{\partial G_1}}{v^2} \tag{A.3}$$

where $u = G_1$ and $v = 1+G_1G_2$. Then equation (A.3) becomes

$$\frac{\partial T}{\partial G_1} = \frac{(1+G_1G_2).1 - G_1G_2}{(1+G_1G_2)^2} = \frac{1}{(1+G_1G_2)^2} \tag{A.4}$$

Then substituting for $\frac{\partial T}{\partial G_1}$ and T in equation (A.2) using, respectively, equations (A.4) and (A.1) yields

$$S_{G_1}^T = \frac{1}{\left(1+G_1G_2\right)^2} \frac{G_1}{\frac{G_1}{1+G_1G_2}} = \frac{1}{1+G_1G_2} \tag{A.5}$$

The sensitivity equation (A.5) shows that the sensitivity function can be made as small as we desire by increasing the return difference $(1+G_1G_2)$.

The foregoing sensitivity theory may be applied to linear dynamic systems simply by replacing the constants, G_1 and G_2, by transfer functions, $G_1(s)$ and $G_2(s)$. Then the sensitivity function (A5) becomes

$$S_{G_1}^T(s) = \frac{1}{1+G_1(s)G_2(s)} \tag{A.6}$$

The associate transfer function,

$$R_{G_1}^T(s) = \frac{G_1(s)G_2(s)}{1+G_1(s)G_2(s)} \tag{A.7}$$

is referred to as the complementary sensitivity function since

$$S_{G_1}^T(s) + R_{G_1}^T(s) = 1 \tag{A.8}$$

Note that viewed in the frequency domain, the effect of the feedback control system in reducing the sensitivity is achieved for those frequencies where $|1+G_1(j\omega)G_2(j\omega)|$ is large.

A.2Computer Aided Pole Assignment

A.2.1 Background

Instability has been experienced when attempting OBRC for some plants, despite the observer correction loop being stable when isolated from the rest of the system. Since the plant model in the observer is mismatched with respect to the real plant and the gains of the correction loop cannot be infinite, insertion of the observer in the system will shift the poles of its model correction loop and the poles of the model state feedback control loop will not be in the originally planned locations. The instability is attributed to this process. It was therefore decided to attempt design of the OBRC system by pole placement using the combined sets of observer and state feedback gains. This, however, proved to be very time consuming due to the complex block diagram algebra if the order of the plant exceeds two and even becomes near impossible in some cases. This provided the motivation to develop a numerical method for such pole assignment and a Matlab script for its automation. Two schemes were investigated initially, one in which the coefficients of the characteristic equation have to be linear with respect to the gains and the other free of this restriction in order to cater for products of the observer and state feedback gains that appear in the characteristic equation coefficients of the complete OBRC system. The restricted version, has proven to be successful and this presented in the following sections. The need for a computer implemented tool was originated by the author. The solution was produced by S J Dodds and the author prepared the MATLAB code listed in section A2.4. Further research is necessary to derive an unrestricted version.

A2.2 Linear Characteristic Polynomial Interpolation

In many linear systems that can be designed by pole assignment, the characteristic polynomial

$$s^n + a_{n-1}(\mathbf{k})s^{n-1} + \cdots + a_1(\mathbf{k})s + a_0(\mathbf{k})$$

has coefficients that are linear functions of the adjustable parameters, k_1, k_2, …and k_n, such as controller gains or observer gains, where $\mathbf{k} = [k_1, k_2 \ldots k_n]^T$ is a vector formed from these parameters. It is this class of linear system to which the linear characteristic polynomial interpolation (LCPI) method is applicable.

A.2.3 Development of the LCPI Algorithm

The general linear relationship between the gains and the coefficients of the characteristic polynomial can be written as

$$\mathbf{a} = \mathbf{M}\mathbf{k} + \mathbf{a}_0 \tag{A.8}$$

where $\mathbf{a} = [a_0, a_1 \ldots a_{n-1}]^{\mathrm{T}}$ is the vector of coefficients of the characteristic polynomial, $s_n + \sum_{i=0}^{n-1} a_i s^i$, $\mathbf{M} \in \Re^{n \times n}$ is a constant matrix and $\mathbf{a}_0$ is a constant vector. The practical aid needed for the application of the method based on A.8 is the MATLAB-SIMULINK linearisation routine that is normally used to produce a linear state space model of a nonlinear dynamical system about a specified operating point, given its SIMULINK block diagram. Instead, this is used with the block diagram of the linear system or subsystem under development. The resulting linear state space model is then converted to a transfer function. The coefficients of the denominator polynomial are then assembled to form the vector, $\mathbf{a}$. The use of this routine is illustrated in figure A.1.

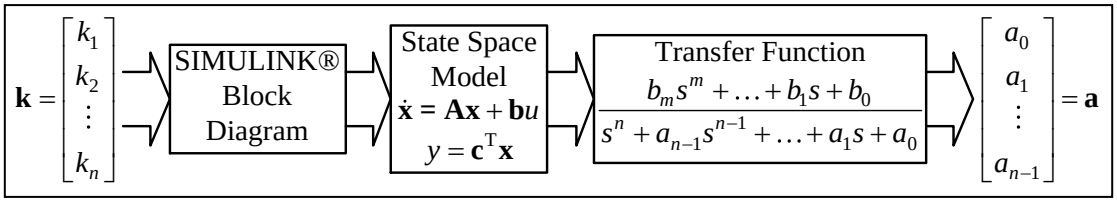


Figure A-1: Computer aided implementation tool for the LCPI method

It is relatively straightforward to determine the desired vector of polynomial coefficients, $\mathbf{a}_{\mathrm{d}}$, to achieve a specified settling time with no overshooting, using the Dodds settling time formulae. For the 5% criterion, the characteristic polynomial is

$$\left[s + \frac{1.5(1+n)}{T_s} \right]^n = s^n + \sum_{i=0}^{n-1} a_{di} s^i \qquad \text{(A.9)}$$

but even these could be determined by a method similar to that illustrated in Fig. A.2 using a SIMULINK block diagram consisting of cascaded identical first order elements as shown in Figure [A-2].

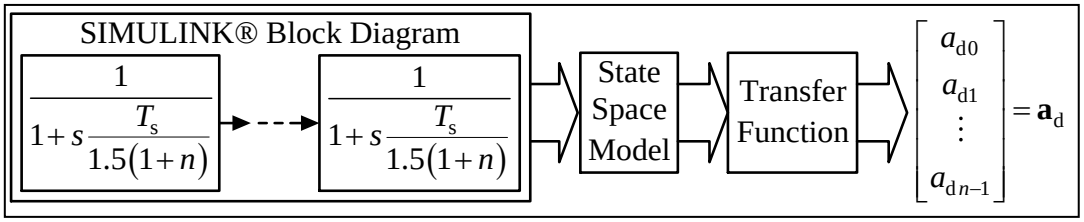


Figure A-2: A computer aided method for calculating the desired characteristic polynomial coefficients

Any other method, however, may be applied. With reference to Figure A-1, the method consists of applying a number of different parameter vectors, $\mathbf{k}_i$, $i = 1, 2, \ldots$, called test parameter vectors, and noting the corresponding coefficient vectors, $\mathbf{a}_i$. $i = 1, 2, \ldots$. Then this information is used to determine the desired value of $\mathbf{k}$, denoted $\mathbf{k}_\mathrm{d}$, that yields $\mathbf{a} = \mathbf{a}_\mathrm{d}$ by multivariable linear interpolation. As will be seen, the minimum number of parameter vectors is $(n+1)$. Linear regression using more data than this is unnecessary because no random errors are involved. Once $\mathbf{M}$ and $\mathbf{a}_0$ have been determined, then, since that (A.8) is satisfied by $\mathbf{a} = \mathbf{a}_\mathrm{d}$ and $\mathbf{k} = \mathbf{k}_\mathrm{d}$ the required parameter vector, $\mathbf{k}_\mathrm{d}$, is obtained as follows

$$\mathbf{a}_\mathrm{d} = \mathbf{M}\mathbf{k}_\mathrm{d} + \mathbf{a}_0 \Rightarrow \mathbf{k}_\mathrm{d} = \mathbf{M}^{-1}\left[\mathbf{a}_\mathrm{d} - \mathbf{a}_0\right] \qquad \text{(A.10)}$$

It is evident from (A.8) that $\mathbf{a}_0$, is the value of $\mathbf{a}$ with $\mathbf{k} = \mathbf{0}$. The computer aided implementation tool of Figure A.1 will therefore yield $\mathbf{a} = \mathbf{a}_0$ when the input is $\mathbf{k} = \mathbf{0}$. To determine $\mathbf{M}$, let $(n+1)$ test parameter vectors, $\mathbf{k}_{t1}$, $\mathbf{k}_{t2}$, …, $\mathbf{k}_{tn+1}$ be chosen (how being determined shortly), then applied, one at a time, to the tool of Fig. A.1 and the corresponding coefficient vectors, $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{n+1}$, noted. To utilise this data, (A.1) may be written down for each pair, $(\mathbf{k}_{ti}, \mathbf{a}_i)$.

Thus,

$$\mathbf{a}_i = \mathbf{M}\mathbf{k}_{ti} + \mathbf{a}_0,\ i = 1, 2, \ldots, n+1 \tag{A.11}$$

Now $\mathbf{a}_0$ may be eliminated between consecutive equation pairs taken from (A.11) by subtracting one from the other, as follows.

$$\mathbf{a}_{i+1} - \mathbf{a}_i = \mathbf{M}[\mathbf{k}_{ti+1} - \mathbf{k}_{ti}],\ i = 1, 2, \ldots, n \tag{A.12}$$

This may be written as

$$\Delta\mathbf{a}_i = \mathbf{M}\Delta\mathbf{k}_{ti},\ i = 1, 2, \ldots, n \tag{A.13}$$

Now (A.13) may be written as a single matrix equation by assembling the coefficient difference vectors, $\Delta\mathbf{a}_i$, and the corresponding parameter difference vectors, $\Delta\mathbf{k}_{ti}$, as the columns of coefficient difference and parameter difference matrices, $\Delta\mathbf{A}$ and $\Delta\mathbf{K}_t$, as follows.

$$\underbrace{[\Delta\mathbf{a}_1 \,|\, \Delta\mathbf{a}_2 \,|\, \ldots \,|\, \Delta\mathbf{a}_n]}_{\Delta\mathbf{a}} = \underbrace{[\Delta\mathbf{k}_{t1} \,|\, \Delta\mathbf{k}_{t2} \,|\, \ldots \,|\, \Delta\mathbf{k}_{tn}]}_{\Delta\mathbf{K}_t} \tag{A.14}$$

Since the test vectors, $\Delta\mathbf{k}_{ti}$, have n elements, the matrix, $\Delta\mathbf{K}_t$, is square and therefore the matrix, $\mathbf{M}$, may be determined as

$$\mathbf{M} = \Delta\mathbf{A}[\Delta\mathbf{K}_t]^{-1} \tag{A.15}$$

provided $\Delta\mathbf{K}_t$ is non-singular. An arbitrary set of test parameter vectors could be chosen within the restriction of non-singularity of $\Delta\mathbf{K}_t$ but to guarantee numerical accuracy it is essential for $\Delta\mathbf{K}_t$ to be well conditioned, i.e., to have a relatively small condition number, close to unity in the range $[1, \infty]$, defined as $|\lambda_{mar}| / |\lambda_{\min}|$, where λ_{mar} and $\lambda_{\min}$ are, respectively, the eigenvalues of $\Delta\mathbf{K}_t$ having the maximum and minimum magnitudes. An ideal choice is

$$\Delta\mathbf{K}_t = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \lambda \end{bmatrix} \tag{A.16}$$

where λ is a real, non-zero constant. This gives the smallest possible condition number (i.e., unity) as all the n eigenvalues are equal to λ and also avoids the matrix inverse in (A.15) by replacing it with a scalar division by λ. Thus

$$\mathbf{M} = \frac{1}{\lambda}\mathbf{\Delta A} \tag{A.17}$$

Let the columns of (A.16) be written

$$\boldsymbol{\lambda}_1 = \begin{bmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \boldsymbol{\lambda}_2 = \begin{bmatrix} 0 \\ \lambda \\ \vdots \\ 0 \end{bmatrix}, \ldots, \boldsymbol{\lambda}_n = \begin{bmatrix} 0 \\ \vdots \\ \lambda \end{bmatrix} \tag{A.18}$$

Then in view of (A.12), (A.13), (A.14) and (A.16)

$$\Delta\mathbf{k}_{ti} = \mathbf{k}_{ti+1} - \mathbf{k}_{ti} = \boldsymbol{\lambda}_i, \quad i = 1, 2, \ldots\ldots, n \tag{A.19}$$

At the beginning, only one test parameter vector has to be chosen. Let this be $\mathbf{k}_{t1}$. Then the remaining n test parameter vectors can be found from (A.19). Thus

$$\mathbf{k}_{ti+1} = \mathbf{k}_{ti} + \boldsymbol{\lambda}_i, \ i = 1, 2, \ldots\ldots, n \tag{A.20}$$

It remains to consider the choice of λ. It is not actually critical and the calculations are even simpler by letting $\lambda = 1$ since (A.17) becomes just

$$\mathbf{M} = \mathbf{\Delta A} \tag{A.21}$$

A.2.4 Summary of the steps of the numerical pole placement procedure

- **Step 1:-** Create a SIMULINK block diagram of any system having the desired closed loop characteristic equation such as in Fig. 10.4
- **Step 2:-** Initialised the coefficient $K_n = 0$
- **Step 3:-** and then run the following command in Matlab to get the characteristics equation of the system by using 'linmod', which will give you the state space equation [A B C D]=linmode(<model name>)
- **Step 4:-** Use the 'poly' to get characteristic polynomial coefficients $a_i = poly(A)$
- **Step 5:-** Form the columns, λ and form the next set of coefficient values for $K_{n+1} = K_n + \lambda_n$ go to step 3
- **Step 6:-** Calculate the coefficient difference between vectors, $\Delta a_{ti} = a_1, i = 1, 2, \ldots\ldots, n$ from the polynomial
- **Step 7:-** Calculate the vector of desired gains using $K_d = [\Delta A^{-1}[a_d - a_0]]$
- **Step 8:-** Now run the Step 3 by using K_d and the transfer function should match the desired closed loop system transfer function

A.2.5 Matlab Script

```
%%
%
%%
% The imput to this M-file are:
%    1:- Model Name
%    2:- Observer Correction Gain or Controller Gain
%    3:- Number of integrator
%    4:- Observer correction order system order Plus one
%    4:- Settling time Ts
%    5:- Settling time of the filter Tso
%
%%
% This function calulates
%    gain of the controller
%    Gain of the Observer
%    DC gain of the system
%
%%
%%
function [cgain dcg]=,...
CalConObsGain(modelname,gain,plantorder,Ts,Tso)
%
% Initialis the varaibles
%
s = tf('s');
alpha=[];
gmat=[];
difgmat=[];
xdifAlpha=[];
```

```
% the overall system shall be observer order which is Plant
or +1;
% the over all system shall be observer order + Plant order
% the obevser
Obserorder=plantorder+1;

Sysorder=Obserorder+plantorder;

% inpk=zeros(norder,1);
inpk=zeros(Sys_order,1);
% create gain difference Matrix
mzero=zeros(Sys_order,Sys_order);

% determine the reference input polynomial coefficients
% since this is associated witht he observer
% the coefficent are calulate
% dusing the observer settling time.
Gob=(s+1.5*(Obser_order+1)/Tso)^Obser_order;

in_coeff= tfdata(Gob,'v');

%reverse the order of the cofficient fro S^0 to S^n-1.
Input_coeff=in_coeff(length(in_coeff):-1:2);

%%

for i=1:Sys_order;
     mzero(i,i)=1;
end
%gmat=rand(norder+1,norder)';
% replace the above with the difference matrix
```

```
gmat=mzero;
for i=1:Sys_order+1;
if (i==1)
     testgain= zeros(Sys_order,1);
else
     testgain =gmat(:,i-1);
end
inpk=inpk+testgain;
inpk_r= [inpk; Input_coeff'];
     assignin('base',gain,inpk_r);
[A,B,C,D] = linmod(modelname);
res=poly(A);
alpha = [alpha res(Sys_order+1:-1:2)'];

end

%% form the difference of the Kmatrix
% the gain is the difference because
% initail matrix to and using unit matix
for i=1:Sys_order;
difAlpha(:,i)=[alpha(:,i+1)-alpha(:,i)];
end
%desire characteristequ.
Cs=(s+1.5*(plantorder+1)/Ts)^plantorder;
Go=(s+1.5*(Obser_order+1)/Tso)^Obser_order;
Gds=Cs*Go;
dgain=tfdata(Gds,'v')';

desgain=dgain(length(dgain):-1:2);
%ensure that the desire gain is a column vector
[m n] =size(desgain);
```

```
if (m<n)
     desgain=desgain';
end
% determin the Q matrix
cgain= difAlpha^-1 *(desgain(:,1)-alpha(:,1));
cgain=[cgain;Input_coeff'];
assignin('base',gain,cgain);
[A,B,C,D] = linmod(modelname);
sys=tf(ss(A,B,C,D));
dcg=dcgain(sys);
end
```

A.2.6 Polynomial Coefficients

```
%%
function [Gob fs hs Is dcg]=,...
DeterminLCPIGainVar(TranferFunction,Ts,Tso,IntgTerm,varargi
n)
%
% Dtermine the closed loop gains for the Observer and the
cntroller
% Input to this module is
%    Transfer function
%    Settling time for the observer and the controller
%%
% Output of this M-file will be
% Controller gain#
% Observer Gains
% correction for the miss match

% make the S as variable
global MinOrder;
s = tf('s');
% determine the order of the system
%%
% This routine provides three options.
%%
% Option 1:
%%
% The default setting is to the split the poles
% between the Ts and Tso.
%%
% Option 2:
%%
```

```
% The User can choose the number of poles
% to be placed at Ts and the
% remianing poles will be placed at Tso.
%%
% Option 3:
%%
% The user will provide a polynominal and
% the remain poles will be palced
% at the TS
%
%% Option 4:
%
% This is the combination of the three
%
%%
% Initialised user define parameter

%%
% Check the additional agreement.
%[Ts : Characteristic equation]
NumberParameterAllow = 6;
NumberOfParameters=nargin;
UseDefaultsetting = 1;
if (NumberOfParameters > NumberParameterAllow)
     disp('Too Many paramter');
     return
end

%
Tfplant=TranferFunction;%*(s^(IntgTerm));
[gnum1 gdnum1]=tfdata(Tfplant,'v');
```

```
% Need to ensure the transfer function in standard
% form eg the highest
% power of denomintor is unit.
gnum=gnum1/gdnum1(1);
gdnum=gdnum1/gdnum1(1);
plant_tf=tf(gnum,gdnum);

%Get the Plant order
%%
Porder=length(gdnum)-1;
% assuming the order of the numerator is the same or less

%for the martrix for the d term multiplier
%%
% The overall system will be the plant
% order + observer order (Porder+1);
%
Oborder=Porder;
ClsySOrder=Oborder+Porder;
ClsysCoeff=ClsySOrder+1;
%
% Created the plant order * observer order matrix.
%%
     PMat= zeros(ClsySOrder,ClsySOrder);
     STermsNumMat= zeros(ClsySOrder,1);
     STermsDenMat= zeros(ClsySOrder,1);
     GainCoeff =zeros(ClsySOrder,1);

% form the cofficient of the S term which
% of numerator and denomator
%%
```

```
DenIndexOffset=1;
for i=1:length(gnum)-1;
     STermsNumMat(i,1)=gnum(i+DenIndexOffset);
end
for i=1:length(gdnum)-1;
     STermsDenMat(i,1)=gdnum(i+DenIndexOffset);
end

% Fill in the matrix with the coefficient of
% the denuminator and numerator.
%%
     % in the n
%if (IntgTerm)

for i=1:Oborder;
     ShiftRowIndex=i-1;
     for j=1:Oborder+1;
          PMat(j+ShiftRowIndex,i)=gdnum(j);
     end
end


 ColIndexOffset=Oborder;
for i=1:Porder+1;
  ShiftColIndex=i-1;
  for j=1:Porder+1;
   PMat(j+ShiftColIndex,i+ColIndexOffset)=gnum(j);
  end
end

% determine the reference input polynomial coefficients
% since this is associated witht he observer
```

```
% the coefficent are calulate
% using the observer settling time withdesire equation.
%%
Gob=(s+1.5*(Oborder+1)/Tso)^Oborder;
%in here we are reversing the data. in_coeff(1)=r0 and so
forth
in_coeff= tfdata(Gob,'v');


%reverse the order of the cofficient fro S^0 to S^n-1.
RCoeff=in_coeff(length(in_coeff):-1:2);
%
RCoeff=1;
%%;
%desire characteristequ.
%%


DesirDnum =,...
DisredCharacteristicEquat(Ts,Tso,Porder,Oborder,varargin);


% Lets form the the C=Pg
%%
desireCoeff=DesirDnum(2:length(DesirDnum))';
if (MinOrder)
     desireCoeff=DesirDnum';
     CTerm=desireCoeff;
else
desireCoeff=DesirDnum(2:length(DesirDnum))';
CTerm=desireCoeff-(STermsDenMat+STermsNumMat);
```

```
end
% Calculate the feed back gains. of the controller
GainCoeff=PMat\CTerm


% To form the closed loop modelis
%%
if (MinOrder)
     hden=[GainCoeff(Oborder+1:length(GainCoeff)-
     IntgTerm)'];
     fden=[GainCoeff(1:Oborder)'];

     if(IntgTerm == 0)
          Intg=ones(1,IntgTerm+1);
          Icoeff=ones(1,IntgTerm+1);
     else
          Icoeff=[GainCoeff(Oborder+Porder+1:length(GainCoe
          ff))'];
          Intg=zeros(1,IntgTerm+1);
          for i=1:IntgTerm
               Intg(i)=1;
          end

     end
else
     hden=[1 GainCoeff(Oborder+1:length(GainCoeff))'];
     fden=[1 GainCoeff(1:Oborder)'];
```

```
     Intg=1;
     Icoeff=[GainCoeff(Oborder+Porder+1:length(GainCoeff))'
     ];
  end
  fs= tf([1],fden);
  hs=tf(hden,1);


   Is=tf(Icoeff ,Intg);
   Gcl= (plant_tf*fs*Is)/(1+fs*hs*plant_tf+ fs*
   plant_tf*Is);


% assignin('base',GainVectorStrName,cgain);
% [A,B,C,D] = linmodv5(modelname);
% sys=tf(ss(A,B,C,D));
dcg=1/dcgain(Gcl);
end
%%
% return the desired Transfer function
%%
function DessiredChEqu=,...
DisredCharacteristicEquat(ts,tso,porder,oborder,varargin)
global MinOrder;
s=tf('s');
if (MinOrder)
     porder=porder-1;
     oborder=oborder-1;
end
```

```
NumberOfParameters=length(varargin{:});
switch (length(varargin{:}))
     case 0
        Gu=1;

case 1
     UserParOne=Get_input_Field(NumberOfParameters,1,vararg
     in);
     [n m]=size(UserParOne);
     if (n>1 || m > 1)
          NumberOfPole= length(UserParOne)-1;
          if (NumberOfPole < oborder)
               oborder= oborder-NumberOfPole;
               Gu=tf(UserParOne,[1]);
          elseif (NumberOfPole < (oborder+podered))
               oborder=0
               porder= (porder+oborder)-NumberOfPole;
               Gu=tf(UserParOne,[1]);
          else
               oborder=0;
               porder=0;
               Gu=1;
          end
      elseif(UserParOne <= porder)

               porder=porder-UserParOne;
               oborder=oborder+UserParOne;
               Gu=1;
          else
               Gu=1;
               oborder=0
               porder=0;
```

```
end
     case 2
          UserParOne=Get_input_Field(NumberOfParameters,1,v
          arargin);
          UserParTwo=Get_input_Field(NumberOfParameters,2,v
          arargin);
          oborder=UserParTwo;
          porder=UserParOne;
          Gu=1;
%              NumberOfPole= length(UserParTwo)-1;
%              TotalUserDefinePOles=
               NumberOfPole+UserParOne;
%              if (TotalUserDefinePOles > oborder)
%                   oborder=0.
%                   porder=0;
%                   Gu=1;
%                   if((TotalUserDefinePOles-
                    oborder)<porder)
%                        porder=porder-
                         (TotalUserDefinePOles-oborder);
%                   end
%              else
%                   oborder=oborder-(TotalUserDefinePOles);
%                   porder=porder+UserParOne;
%                   Gu=tf(UserParTwo,[1]);
%              end
          case default
               oborder=0.
               porder=0;
               Gu=1;
end
```

```
Go=(s+1.5*(oborder+1)/tso)^(oborder);
Cs=(s+1.5*(porder+1)/ts)^porder;
Gds=1/(Go*Cs*Gu);
[DesireNum DessiredChEqu]=tfdata(Gds,'v');

end


%%
% Gets the parameters from the input arguments vector.
%
function
input_para=Get_input_Field(nb_parameter,index,varargin)
     if(index<=nb_parameter)
          cell_array=varargin{:};
          inpara=cell_array{1};
          input_para=inpara{1,index};
     else
          input_para = '';
     end
     return;
end
```

REFERENCES

[1] A. H. C. D. A. H. E. J. BARNETT, CControl Systems and Control Engineering Estimators and Observers, Wiki Web, sep 2007.

[2] P. G. BEALE, Lecture notes, http://teal.gmu.edu/ gbeale/examples.html, 2000.

[3] A. K. D.C.MCFARLANE, Robust Controller Design Using Normalised Coprime Factor Plant Description., Springer-Verlag Lecture Notes in Contytrol and Information Sciences, jun 1990.

[4] S. J. DODDS, System with different observer design methods, The International Control Conference..

[5] S. J. DODDS, A novel approach to robust motion control, 5th International Conference, 2004.

[6] J. D. B. A. T. A. FRANCIS, Feedback ControlTheory, Macmillan Pubishing Co, 1990.

[7] M. A. M. B. P. B. HAJER GHARSALLAOUI, Flatnessbased Control and Conventional RST Polynomial Control of a Thermal Process, Int.J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Sept. 2009.

[8] M. HELLSTR´LOM, Engine speed based estimation of the. Master's thesis, Dept. of Electrical Engineering, Link´lopings universitet, Feb 2005.

[9] IEEE., State-space solution to standard and control problemvol. Vol.34, IEEE, Aug 1989.

[10] P. G. Z. P. IOAN D. LANDAU, Digital Control System Desing, Identification. chapter3, 2006..

[11] A. G. K. J.A.SEFTON, Poles-zero cancellation in general problem with refenceto

two block design, System and control letters.

[12] H. E. A. K. T. R. JENKINS, Adaptive pole-zero cancellation in grinding, force control. http://www.deltatau.com/common/support/appex/AllAppex/pmrc5.pdf, Sept 2005..

[13] C. B. B. JOSEPH, Techniques of Model-Based Control Chapter 2, Prentice Hall, 2002.

[14] S. A. P. M. KAITWANIDVILAI, A genetic-algorithm-based fixedstructure robust H−∞, SAE Technique paper series, 2002.

[15] A. G. A. C. M.-S. KALMUKALE, Data-based internal model controller design for a class, European Symposium on Computer Arded Process Engineering (2005)., 2005.

[16] A. KARIMI, Robust control, ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE, 2005.

[17] A. LAGERBERG, Control of backlash in automotive powertrains, PhD thesis, hamers university Sweden, 2004.

[18] A. A. S. B. LAGERBERG, Control of backlash in automotive powertrains, IEEE Conference , 2009.

[19] I. LANDAU, The R-S-T digital controller design and applications, Control Engineering Practice 6 1998 155, Dec 1998.

[20] F. S. LEWIS, observer and regulator design, Paper, 2001.

[21] A. F. A. S. S. E. LYNCH, Internal model-based controller design using, paper.

[22] I. THE MATHWORKS, Control Design Course, Sminar version 1.1, Dec 2000.

[23] I. THE MATHWORKS, Embedded Torque Estimator for (2004), SAE., SAE., 2004.

[24] J. J. A. H. P. C. MOSKWA, Engine load torque estimation using vol. 0-7803-2685-719, IEEE, Proceedings of the 34th.

[25] Q. C. Y. L. H. E. J. NIU, H-infinity control theory in vehicle diesel fuel injection pump control system design fuel injection systrem, Paper on the Web, Dec 2010.

[26] U. O. E. LONDON, Observer Based Robust Control, Proceedings of Advances in Computing and Technology, 2007.

[27] J. Kobe, LINEAR DRIVES FOR INDUSTRY APPLICATIONS. Mode of a Vacuum Air Bearing Linear Drive, 2005.

[28] E. A. G. E. OSTERTAG, RST-Controller Design for Sinewave References byMeans of an Auxiliary Diophantine Equation, 44th IEEE Conference, Dec 2005.

[29] E. A. G. E. OSTERTAG, Rst-conttroller design a ratinal teaching mehtod based on two diophantine equation, Control Engineering Practice , Dec 2006.

[30] G. F. A. G. L. PAOLO FALCONE, Non-linear net engine torque estimator, http://web.ing.unisannio.it/fiengo/Download/Pers/Torque, 2005.

[31] D. SIMON, Analyzing control system robustness, sept 2008.

[32] A. R. S.J.WILLIAMS, A comparison of different h-infinity design methods for vstol flight control system design, San Diego: In American Control Conference pp. 2508–2513, May 1990.

[33] S. A. P. I. SKOGESTAD, Multivariable feedback Control Analysis and design, Jonn Wiley and Sons, 1993.

[34] N. S.NISE, Control Systems Engineerin, 2nd edition Chapter7.7, Road,Menlo

Park,CA: Addison-Welsely, 2725 Sand Hill, 1995.

[35] S.Dodds, Settling time formulae for the design of control systems with linear closed loop dynamics, UNIVERSITY OF EAST LONDON., 2008.

[36] G. VAN DER ZALM, DEVELOPMENT OF A TUNING METHOD FOR SPEED CONTROLLERSIN DAF TRUCKS, DAF-report nr. 51051/04-049 , June 2004.

[37] J. A. D. S. J. VITTEK, Forced Dynamic Control of Electric Drives, University of Zilina Press,book ISBN 80-8070-087-7, 2003.

[38] G. A. B. G. WELCH, An Introduction to the Kalman Filter'. University of, Chapel Hil : University of North Carolina , 2005.

[39] P. J. Maloney, Embedded Torque Estimator for Diesel Engine Control Application, SAE International, 2004 .

[40] M. A. F. K. G. Javier Francoa, Real-time brake torque estimation for internal combustion engines, Mechanical Systems and Signal Processing 22 338–361, 2008.

[41] J. e. a. Doyle, State-Space Solutions to Standard HS and H-infinity Control Problems, IEEE Transactions on Automatic Control 34.8 , 1990.

[42] J. K. G. P. K. a. B. F. Doyle, State-Space Solutions to Standard HS and H-infinity Control Problems, IEEE Transactions on Automatic Control 34, no. 8, 1989.

[43] B. F. A. T. John Doyle, Feedback Control Theory, Macmillan Publishing co, 1990.

[44] D. E. Rivera, Internal Model Control, College of Engineering and Applied Sciences Arizona State University Tempe, Arizona 85287-6006.

[45] D. M. M. a. S. S. Rivera, Internal Model Control 4. PID Controller, Ind. Eng. Chem. Process Des. Dev. 25, 252, , 1986.

[46] M. S. a. M. Lee, IMC Filter Design for PID Controller Tuning of Time Delayed Processes, 2012.

[47] M. a. E. Z. Morari, Robust Process Control, NY: Prentice-Hall, Englewood Cliffs, 1989.

[48] M. T. Tham, “Internal model control,” Chemical and Process Eng, University of Newcaslte upon Tyne, 2002.

[49] Y. H. D. ZWEIRI, “engine indicated torque estimation based,” vol. VOLUME 1, June 2006.

[50] R.F.HARRISON, “Robust and Mutivariable Control,” Msc Lecuture notes of University Of Sheffield , Sept 2000.

[51] D. SIMON, “H-infinity filtering with inequality constraints for aircraft turbofan engine health estimation,” ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE, Sept 2005.

[52] J. D. C. J. S. M. V. P. SAS, “H-infinity feedback control for signal tracking on a 4 poster testrig in the automotive industry,” 2002.

[53] G. Rizzoni, “Estimate of IC Engine Torque from Measurement of Crankshaft Angular Position,” Sept 1993.

[54] B. Lee, G. Rizzoni, Y. Guezennec and A. Soliman, Engine Control Using Torque Estimation, SAE TECHNICAL, 2001.

[55] B. LEE, G. Y. G. RIZZONI and A. S. A. , “Engine control using torque estimation.,” 2001.

[56] R. N. CLARK, “Another reason to eschew pole-zero,” April 988.

[57] K. ASTROM and B. WITTENMARK, Control system Theory and design chapter

10, NY: Parentice-Hall.

[58] A. K and B. WITTENMARK, Control system Theory and design, Computers Communications & Control, 2009.

[59] B. S.W.S.J and P. L. LEE, “Passivity based imc control for multivariable nonlinear,” 2005.

[60] A. H. BARNETT, M. BLAAUBOER, A. MODY and E. J. AND HELLER, “dynamics of air-breathing hypersonic. Flight path angle,” paper, 2004.

[61] Y. H. Zweiri, “Diesel Engine Indicated Torque Estimation Based on Artificial Neural Networks,” no. 1.

[62] K.J.HUNT, R.P.JAIME and H.GOLLEE., “Robust control of electrically-Stimulated Muscle using Polnomial H_infinity design,” University of Glasgow, 2004.

[63] X.-D. SUN, P. G. Scots. and G. BALFOUR, “Application of loop shaping H-infinity control to diesel engine control,” Mar 2002.